

NUMERICAL METHODS FOR PARASITIC EXTRACTION OF ADVANCED
INTEGRATED CIRCUITS

A Dissertation

by

YUHAN ZHOU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Weiping Shi
Committee Members,	Robert D. Nevels
	Peng Li
	Vivek Sarin
Head of Department,	Miroslav M. Begovic

December 2017

Major Subject: Computer Engineering

Copyright 2017 Yuhan Zhou

ABSTRACT

FinFETs, also known as Fin Field Effect Transistors, are a type of non-planar transistors used in the modern integrated circuits. Fast and accurate parasitic capacitance and resistance extraction is crucial in the design and verification of FinFET integrated circuits. Though there are wide varieties of techniques available for parasitic extraction, FinFETs still pose tremendous challenges due to the complex geometries and user model of FinFETs. In this thesis, we propose three practical techniques for parasitic extraction of FinFET integrated circuits.

The first technique we propose is to solve the dilemma that foundries and IP vendors face to protect the sensitive information which is prerequisite for accurate parasitic extraction. We propose an innovative solution to the challenge, by building a macro model around any region in 2D/3D on a circuit where foundries or IP vendors wish to hide information, yet the macro model allows accurate capacitance extraction inside and outside of the region.

The second technique we present is to reduce the truncation error introduced by the traditional Neumann boundary condition. We make a fundamental contribution to the theory of field solvers by proposing a class of absorbing boundary conditions, which when placed on the boundary of the numerical region, will act as if the region extends to infinity. As a result, we can significantly reduce the size of the numerical region, which in turn reduces the run time without sacrificing accuracy.

Finally, we improve the accuracy and efficiency of resistance extraction for FinFET with non-orthogonal resistivity interface through FVM and IFEM. The performance of FVM is comparable to FEM but with better stability since the conservation law is guaranteed. The IFEM is even better in both efficiency and mesh generation

cost than other methods, including FDM, FEM and FVM.

The proposed methods are based on rigorous mathematical derivations and verified through experimental results on practical examples.

DEDICATION

*To my husband, Wei Zhang,
for being with me and experience everything together.*

*To my parents and relatives,
for their unconditional love and support.*

*To my colleagues and friends,
for their discussions and insights through my study.*

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Weiping Shi, for his endless support, encouragement, and guidance throughout the course of my Ph.D. study. I appreciate that he gladly accepted me to take my first step for becoming a researcher under his guidance. His patience allowed me to make steady and consistent progress. Whenever I was stuck on a problem, he has been always with me to give precious advice to help me moving forward. Without his help, I would not have completed this work.

I would also like to thank my committee members, Dr. Robert D. Nevels, Dr. Peng Li, and Dr. Vivek Sarin for their insightful comments and support. Without the knowledge I learned from their classes and their constructive comments, this work would not have been possible.

My thanks also go to Dr. Daren B.H. Cline, Dr. Raytcho Lazarov, Dr. Alan Demlow, Dr. Zhilin Li, Dr. Ivar Aavatsmark, Dr. Tim Davis, Dr. Daniel Ioan for gladly taking their time for discussion and collaboration and making my life pleasurable.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis (or) dissertation committee consisting of Professor Weiping Shi, Peng Li and Robert D. Nevels of the Department of Electrical and Computer Engineering, and Professor Vivek Sarin of the Department of Computer Science and Engineering.

All work for the dissertation was completed by the student, under the advisement of Weiping Shi of the Department of Electrical and Computer Engineering, and in collaboration with Yong Zhang of the Department of Electrical and Computer Engineering and Daren B.H. Cline of the Department of Statistics.

Funding Sources

Graduate study was completed without any outside financial support.

NOMENCLATURE

ABC	Absorbing Boundary Condition
AL	Artificial Layer
FDM	Finite Difference Method
FEM	Finite Element Method
IFEM	Immersed Finite Element Method
FVM	Finite Volume Method
FRW	Floating Random Walk
TPFA	Two-point Flux Approximation
MPFA	Multi-point Flux Approximation
DOF	Degree of Freedom
PEBI	Perpendicular Bisector
LGR	Local Grid Refinement
SOI	Silicon On Insulator

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iv
ACKNOWLEDGMENTS	v
CONTRIBUTORS AND FUNDING SOURCES	vi
NOMENCLATURE	vii
TABLE OF CONTENTS	viii
LIST OF FIGURES	x
LIST OF TABLES	xiv
1. INTRODUCTION	1
1.1 Background	1
1.2 Overview of Our Work	1
1.3 Outline	5
2. MACRO MODEL OF ADVANCED DEVICES FOR PARASITIC EX- TRACTION	6
2.1 Background	6
2.2 Extraction flow using Macro Model	8
2.3 Construction of Macro Model for FDM	9
2.4 Using of Macro Model by FDM field Solver	13
2.5 Construction and Using of Macro Model for FEM	16
2.6 Construction and Using of Macro Model for Floating Random Walk Method	18
2.7 Profile Simplification	23
2.7.1 Concept of Equivalent Profile	23
2.7.2 Finding Equivalent Profile	24
2.8 Experimental Results	25
2.8.1 Correctness	25

2.8.2	Robustness	26
2.8.3	Run Time	26
2.8.4	Equivalent Profile Experiment	27
2.9	Conclusion	29
3.	CAPACITANCE EXTRACTION WITH PROVABLY GOOD ABSORB- ING BOUNDARY CONDITIONS	34
3.1	Background	34
3.2	Absorbing Boundary Conditions	37
3.3	Implementation in Cartesian Coordinates	41
3.3.1	Absorbing Boundary Condition (ABC1)	42
3.3.2	Absorbing Boundary Condition (ABC2)	45
3.3.3	Implement ABC1 with Artificial Layer (AL)	50
3.4	Application to other Methods	52
3.4.1	Finite Element Method	53
3.4.2	Floating Random Walk	54
3.5	Experiments	54
3.5.1	Conductors in Free Space	55
3.5.2	Conductors in Multi-layer Dielectric	58
3.5.3	Conductors in SOI	60
3.6	Conclusion	61
4.	RESISTANCE EXTRACTION FOR ANISOTROPIC LAYER	65
4.1	Background	65
4.2	Finite Difference Method	67
4.3	Finite Volume Method	69
4.3.1	Two-Point Flux Approximation	71
4.3.2	Multi-Point Flux Approximation	78
4.4	Finite Element Method	92
4.4.1	Traditional Finite Element Method	92
4.4.2	Immersed Finite Element Method	97
4.5	Simulation and Evaluations	104
4.5.1	Resistance Extraction of non-rectilinear Conductor	104
4.5.2	Resistance Extraction of FinFET Contact	111
4.6	Conclusion	114
5.	CONCLUSION	124
	REFERENCES	126

LIST OF FIGURES

FIGURE		Page
1.1	Traditional Planar transistor and Tri-Gate FinFETs. Reprinted from [1].	2
1.2	FinFET Sub-threshold Slope and Gate Delay Improvements. Reprinted from [1].	3
2.1	Macro model construction and application flow for foundry, EDA vendor and designer.	10
2.2	Demonstration of 2D MOSFET with black box enclosed by dashed lines.	11
2.3	Flow diagram of finding equivalent profile.	30
2.4	3D FinFET Model.	31
2.5	Top view and side view of the 3D FinFET structure and the black box around Fin part.	31
2.6	Top view and side view of the 3D FinFET structure and the black box around Drain part.	32
2.7	Original profile with raised source and drain is shown in (a). Simplified profile is shown in (b). Both profiles produce the same extracted capacitance between device pins and contact.	33
2.8	The original profile of trapezoidal contact is shown in (a), simplified equivalent profile is shown in (b).	33
3.1	Electric potential distribution due to 1 volt conductor above a 0 volt ground plane. The correct distribution computed with a large numerical region is shown in (a). When a small region is used, the traditional Neumann boundary gives inaccurate results as shown in (b), while our absorbing boundary condition gives accurate results as shown in (c).	35
3.2	The naming of neighbors for node P , which is at the $+X$ boundary. Fictitious node R^* is the mirror of node L outside of the $+X$ boundary, and $V(R^*)$ is approximated by ABC, used to compute $V(P)$	41

3.3	Top view of absorbing boundary condition with an artificial layer of thickness d . The boundary of the numerical region is marked with red color while the exterior Dirichlet boundary is marked with purple color.	51
3.4	Experiment I: Two conductors in free space.	55
3.5	Accuracy of total capacitance and coupling capacitance as a function of D for Experiment I shown in Fig. 3.4.	55
3.6	Number of non-zero entries and run time as a function of distance D for Experiment I shown in Fig. 3.4.	56
3.7	Run time as a function of relative error for total capacitance C_{11} in Experiment I.	57
3.8	Experiment II: Part of an inverter in multi-dielectric layers.	58
3.9	Accuracy of total capacitance and coupling capacitance as a function of D for Experiment II shown in Fig. 3.8.	59
3.10	Number of non-zero entries and run time as a function of D for Experiment II shown in Fig. 3.8.	60
3.11	Run time as a function of relative error for coupling capacitance C_{14} in Experiment II.	61
3.12	Experiment III: Three conductors in SOI.	62
3.13	Accuracy of total capacitance and coupling capacitance as a function of D for Experiment III shown in Fig. 3.12.	63
3.14	Run time as a function of relative error for coupling capacitance C_{AB} in Experiment III.	64
4.1	Five point stencil in 2D FDM.	67
4.2	Mesh of CC-FVM and NC-FVM. The unknowns are associated to black nodes, with the control volume marked with shaded domain . .	70
4.3	Two-point flux approximation definition with two cells Ω^i and Ω^{i+1} . .	72
4.4	A Voronoi-Delaunay duality in two dimensions.	74
4.5	Assign fixed Voronoi cell centers to fit a non-orthogonal interface edge.	75

4.6	Mesh generation scheme to fit the non-orthogonal interfaces or boundaries.	76
4.7	Multi-level quad-tree local grid refinement illustration.	77
4.8	Sub-cell Ω_k^i constructed by vertices G_i , $P_{k-\frac{1}{2}}$, P_k and $P_{k+\frac{1}{2}}$	79
4.9	Notation around an internal vertex p_k	86
4.10	Linear interpolation functions for triangle element.	96
4.11	Cartesian triangulation grid and Cartesian grid.	98
4.12	From FEM to IFEM: (a) geometry with non-orthogonal boundary in Cartesian grid; (b) geometry with non-orthogonal boundary in Cartesian triangular grid; (c) Cartesian grid for non-interface element and Cartesian triangular grid for interface element.	99
4.13	Cartesian triangles with an interface line cutting through.	100
4.14	The basis function at node A in Fig.4.13 with IFEM when $\frac{\sigma^+}{\sigma^-} = 10$. .	102
4.15	Experiment I: Resistance extraction case for a very thin non-rectilinear conductor.	105
4.16	Cartesian grid fitting the stripe shaped conductor with different rotation angle α for case in Fig. 4.15. (a) $\alpha = 30^\circ$; (b) $\alpha = 45^\circ$; (c) $\alpha = 60^\circ$	106
4.17	Voronoi grid fitting the stripe shaped conductor with different rotation angle α for case in Fig. 4.15. (a) $\alpha = 30^\circ$ without LGR; (b) $\alpha = 45^\circ$ without LGR; (c) $\alpha = 60^\circ$ without LGR; (d) $\alpha = 30^\circ$ with quad-tree LGR; (e) $\alpha = 45^\circ$ with quad-tree LGR; (f) $\alpha = 60^\circ$ with quad-tree LGR.	116
4.18	Triangular mesh fitting the stripe shaped conductor with different rotation angle α for case in Fig. 4.15. (a) $\alpha = 30^\circ$; (b) $\alpha = 45^\circ$; (c) $\alpha = 60^\circ$	117
4.19	Cartesian triangle with two disjoint interface lines cutting through. .	118
4.20	Cartesian mesh for non-interface elements and Cartesian triangular mesh for interface elements for case in Fig. 4.15. (a) $\alpha = 30^\circ$; (b) $\alpha = 45^\circ$; (c) $\alpha = 60^\circ$. All interface elements are marked with blue color while the rest elements are labeled with yellow color.	119

4.21	Experiment II: Resistivity at the boundary between FinFET gate and contact is anisotropic. (a) different resistivity in the normal direction and tangent direction of the boundary (b) equivalent model with a very thin layer at the boundary.	120
4.22	Cartesian grid fitting the thin layer between FinFET gate and contact in Fig. 4.21.	120
4.23	Voronoi mesh fitting the thin layer between FinFET gate and contact in Fig. 4.21. (a) without local grid refinement (b) with multi-level quad-tree local grid refinement.	121
4.24	Triangular mesh fitting the thin layer between FinFET gate and contact in Fig. 4.21.	121
4.25	Cartesian mesh for non-interface elements and Cartesian triangular mesh for interface elements for case in Fig. 4.21. (a)coarse mesh with two lines cutting through elements (b)dense mesh. All interface elements are marked with blue color. Non-interface elements outside of the thin layer are labeled with yellow color, while non-interface elements insides of the thin layer are labeled with green color.	122
4.26	Accuracy convergence and run time comparison for Experiment II. . .	123
4.27	Run time as a function of relative error for resistance in Experiment II.	123

LIST OF TABLES

TABLE		Page
2.1	2D MOSFET: Comparison of extraction without black box and with different black boxes.	25
2.2	3D FinFET: Comparison of extraction without black box and with different black boxes.	26
2.3	2D MOSFET: Comparison of extraction with different discretization steps inside and outside of black box.	27
2.4	3D FinFET: Comparison of extraction with different discretization steps inside and outside of black box in Fig. 2.6.	28
2.5	Comparison of equivalent profile for MOSFET gate part with respect to adjacent contacts with $\epsilon = 0.1\%$	28
2.6	Comparison of equivalent profile for trapezoidal contact with $\epsilon = 0.1\%$	28
4.1	Feature comparison of FDM, FEM and FVM.	66
4.2	Resistance extraction for Experiment I with FDM.	107
4.3	Resistance extraction for Experiment I with TPFA.	107
4.4	Resistance extraction for Experiment I with MPFA and FEM.	108
4.5	Resistance extraction for Experiment I with IFEM.	111
4.6	Resistance extraction for Experiment II with FDM.	112
4.7	Resistance extraction for Experiment II with TPFA.	113
4.8	Resistance extraction for Experiment II with MPFA and FEM.	114
4.9	Resistance extraction for Experiment II with IFEM.	115

1. INTRODUCTION

1.1 Background

FinFETs, also known as Fin Field Effect Transistors, are a type of non-planar transistors used in the modern integrated circuits. It uses a conducting channel that rises above the level of the insulator, creating a thin silicon structure, shaped like a fin. In a traditional planar transistor, the current between source and drain is closely related to the ratio $\frac{W}{L}$ where W and L are the width and length of the device respectively. As the industry process technology scales down, L is decreased to improve the drive strength of the transistor which results in high susceptibility to variability when leakage control become an important issue to take into consideration. As shown in Fig. 1.1, a tri-gate transistor designed by intel, the gate surrounds the channel on all three sides and has much better control so that all the charge below the transistor is removed. It has better control over the sub-threshold leakage. Since there is no bulk capacitance, which results in small sub-threshold swing. Thus the transistor has faster switching speed. Since the gate delay can be considered as the inverse of the frequency, lower gate delay means faster frequency which is demonstrated in Fig. 1.2. Furthermore, the tri-gate configuration provides a steeper sub-threshold slope (SS) effectively reducing leakage current, thereby decreasing dynamic power consumption and increasing device performance.

1.2 Overview of Our Work

In this dissertation, we solve three practical problems in FinFET integrated circuits.

In order to perform accurate parasitic extraction, foundries must provide the cross-section profile of devices, and IP vendors must provide sufficient layout in-

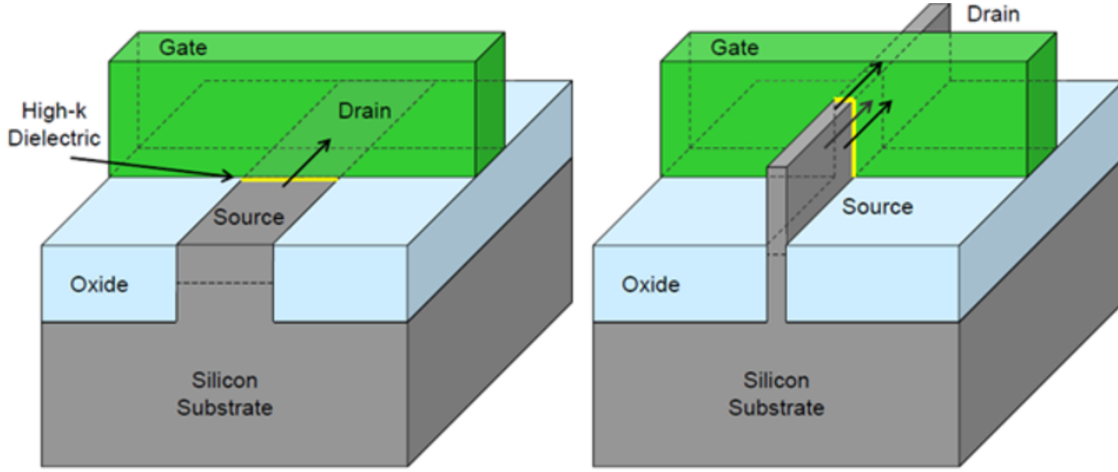


Figure 1.1: Traditional Planar transistor and Tri-Gate FinFETs. Reprinted from [1].

formation. However foundries and IP vendors are increasingly reluctant to reveal such sensitive information, especially for advanced devices, like FinFET. Therefore, the industry is faced with the following challenges: i) foundries/IP vendors need to protect their trade secrets, ii) EDA vendors need to integrate foundry data into extraction tools, and iii) IC designers need to have the accurate parasitic data.

We propose an innovative and practical solution to these challenges, by building a macro model around any region in 2D/3D on a circuit where foundries or IP vendors wish to hide information, yet the macro model allows accurate capacitance extraction inside and outside of the region. We first give algorithms to construct the macro model. Then we describe how existing extraction algorithms can interface with the macro model to perform extraction for the entire circuit at the same accuracy as if complete information were given. The macro model can be used in FDM/FEM and Floating Random Walk (FRW), due to an equivalence theorem we proved. Finally, we propose the concept of equivalent profile, and describe how to find one based on the macro model.

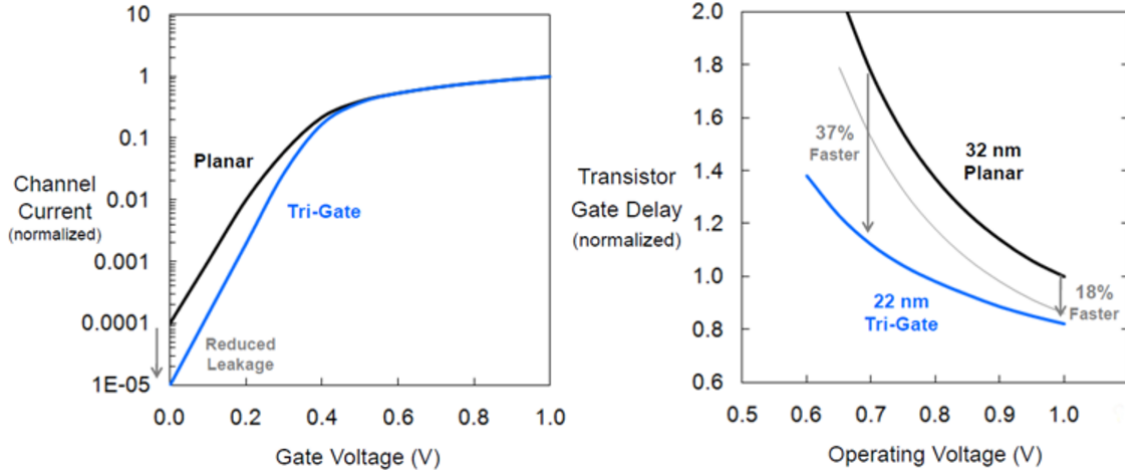


Figure 1.2: FinFET Sub-threshold Slope and Gate Delay Improvements. Reprinted from [1].

Our second project focuses on the error introduced by the Neumann boundary conditions. Due to the increasing 3D effects of FinFET and local interconnect, 3D field solvers become popular for parasitic capacitance extraction of full custom circuits, IPs, and packages. Traditional field solvers based on FDM/FEM/FRW assume Dirichlet or Neumann boundary conditions on the outer boundary of the numerical region. To ensure high accuracy, the numerical region should be large so that the electric field on the boundary will not interfere with the conductors.

We propose a class of absorbing boundary conditions, which when placed on the boundary of the numerical region, will act as if the region extends to infinity. As a result, we can significantly reduce the size of the numerical region, which in turn reduces the run time without sacrificing accuracy. Our absorbing boundary conditions are mathematically derived and proved to be superior than the traditional boundaries and previous absorbing boundaries. In particular, we prove the asymptotic error of the proposed n -th order absorbing boundary is $O(1/r^{n+2})$, while the

error of Neumann boundary is $O(1/r^2)$, where r is the size of the numerical region. Based on our first order absorbing boundary condition, we also propose an equivalent dielectric layer approach that can be easily adopted by any existing field solver. This novel method is effective in improving the run time and accuracy in capacitance problems for multi-layer dielectric and Silicon On Insulator (SOI) cases, over previous boundary conditions for uniform or non-uniform meshes.

The third problem we investigate is doing resistance extraction with non-orthogonal conductivity interface. Many commercial field solvers for resistance extraction are based on finite difference method (FDM) or finite element method (FEM). The linear systems are easy to construct with FDM, but the mesh must be orthogonal. To approximate the non-orthogonal interface, it requires very dense mesh which leads to huge computational cost to achieve high accuracy. On the other hand, FEM is feasible to non-orthogonal interface with the proper mesh fitting the interface. However, it is expensive to construct the local linear system. FEM does not follow the conservation law, which makes it struggle to stability and low accuracy for coarse mesh. Besides, it is not trivial to generate the triangular or quadrilateral mesh for FEM.

We introduce for the first time finite volume method (FVM) for resistance extraction. FVM is similar to FEM which allows non-orthogonal mesh while at the same time, the linear system is easy to construct. In addition, the robustness is comparable to FDM, due to the conservation law is always guaranteed. Our method, a second-order accurate FVM, is based on the latest development in computational physics. Experimental results show that this new scheme is effective in improving the run time and accuracy for resistance extraction. However, both FEM and FVM are faced with the dilemma of unexpected high mesh generation costs. We need to generate new mesh again to adapt to any geometrical change, even it is very trivial.

We also introduced immersed finite element method (IFEM) to solve the dilemma. It can maintain the Cartesian grid and FDM scheme for the regions without non-orthogonal interface or boundary, while applying the extended FEM scheme for the grid cells with interface lines cutting through. It is rather simple and efficient to build local matrix for the cells along the interface line which avoids the heavy burden to build FEM local matrix for all triangle cells. The approach to build local system can be extended to multiple disjoint interface lines cutting through a single cells which can be applied to even coarser rectangle mesh.

1.3 Outline

The major contribution of the dissertation is presenting several novel approaches that resolve the practical problem for the existing parasitic extraction methods in terms of intellectual property, accuracy and adaptability. The rest of the dissertation is organized as follows. Chapter 2 presents the macro model of advanced devices for parasitic extraction. Chapter 3 gives detailed description of the absorbing boundary conditions and the applications. Chapter 4 summarizes numerical methods for resistance extraction including FDM, FEM, FVM and IFEM, and compares the performance for practical problems.

2. MACRO MODEL OF ADVANCED DEVICES FOR PARASITIC EXTRACTION *

In this chapter, we discuss the macro model construction and its applications for the parasitic extraction.

2.1 Background

In order to perform accurate parasitic extraction, foundries must provide cross-section profile information in their PDK (Process Design Kit) to foundry customers. Examples of such information are:

- the height and width of diffusion, field poly and gate poly,
- the shape of conformal dielectric near devices,
- the dielectric constants for gate oxide and spacer,
- the geometry of dummy poly and dummy diffusion,
- the 3D shape of multi-finger FinFET,
- the shape of contact, etc.

LPE (Layout Parasitic Extraction) tools rely on the profile information to accurately perform extraction. Field solvers that are most accurate, such as [2][3][4] all need the profile information to set up Laplacian equations. Rule-based tools, such as [5][6][7], all need the profile information to feed field solvers to generate extraction rules.

*Reprinted from “Macro Model of Advanced Devices for Parasitic Extraction” by Yuhua Zhou, Yong Zhang, Vivek Sarin, Wangqi Qiu, Weiping Shi, 2016. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 35, 1721-1729, Copyright 2016 by IEEE.

However foundries are reluctant to reveal such sensitive information for advanced devices such as FinFET. Foundries are faced with the dilemma of revealing more information at the risk of compromising sensitive trade-secrets, or not revealing enough information and suffer performance loss due to inaccurate extraction. Similarly, IP vendors must reveal layout of IP cells in order for the users to extract parasitic information near the IP cells.

Until now, the foundries use a number of ad hoc approaches to deal with the problem. For example, the foundry may reveal an approximated or altered profile. However, to do so may cause uncontrollable error when extracting capacitance between device pins, such as between source and gate, drain and gate, etc. The tendency of compact model for advanced nodes is to rely on extraction tools to extract fringing parasitic capacitance between device pins, rather than using closed-form expressions. For advanced process, no existing approach seems to work due to the complexity of the geometry near the devices and infinite possibilities by which the device may interact with nearby interconnect.

In this chapter, we solve the dilemma by proposing a macro model for any region in 2D/3D chosen by the foundries or IP vendors, which we call a black box. A macro model accurately describes the electric field relation between the boundary nodes and internal nodes of the black box, without revealing geometrical or profile information inside of the black box. Since the ratio of the number of boundary nodes to internal nodes goes to zero as the mesh size increases, it is theoretically impossible to reverse engineer the detailed information inside of the black box from the macro model. For resistance macro model, the same techniques presented in this chapter can be applied. But for inductance macro model, more study is needed.

2.2 Extraction flow using Macro Model

In [8][9], macro models were built to speedup hierarchical capacitance extraction, where the layout and profile information are fully disclosed. In this chapter, the macro models are built and applied under different conditions, for a different purpose. The concept was first presented by us in [10]. In this section, we will introduce the macro model concept for Finite Difference Method (FDM) first, then describe the capacitance extraction flow using macro model.

A closed region which contains one or more conductors is a *black box* denoted as \mathbb{B} . In general, a black box can be any shape, though a 2D rectangle or 3D cube makes computation easier. The surface of the black box is denoted as \mathfrak{B} . There are uniform or non-uniform discretized mesh on \mathfrak{B} . Those grid points on \mathfrak{B} are called *port nodes*.

In general, when electric potential is applied to each port node separately as vector $\mathbf{V}_{\mathbf{p}}(\mathfrak{B})$, it results in electric flux normal to the Gaussian surface \mathfrak{B} . The electric flux is denoted as $\mathbf{D}_{\mathbf{p}}(\mathbb{B}, \mathbf{V})$.

A Macro model for FDM is a matrix $\mathbf{M}_{p \times p}$, where p is the number of port nodes. Any column k of the matrix represents the flux at each port node when one volt is applied to k th port node and zero volt to all other port nodes. A 2D macro model is described in Def. 1, and 3D modes have similar definition.

Fig. 2.1 is the macro model application flow. The foundry uses the real cross section profile to build the macro models while hiding original profile information. The EDA vendors provide LPE tools that support macro models. Designers will combine the macro models provided by the foundry and LPE tools provided by the EDA vendors to perform extractions for actual circuits.

Definition 1: Macro Model for FDM in 2D

Black Box Region:

$$[x_{min}, x_{max}] \times [y_{min}, y_{max}]$$

Port Nodes:

$$P_k = (x_k, y_k), \quad k = 1, 2, \dots, p.$$

Macro Model Matrix:

$$\mathbf{M}_{p \times p}$$

$$\text{specifying } \mathbf{D}_p = \mathbf{M}\mathbf{V}_p$$

2.3 Construction of Macro Model for FDM

In this section, we present an algorithm to build a macro model. The algorithm is performed at the foundry side. Although the following description and equations are for the 2D case, they are valid for 3D as well. In Fig. 2.2, the boundary of a 2D black-box is indicated by the dashed line.

We discretize the black box with a mesh. For each mesh grid node, we assign a variable representing the potential at that node. There are three types of nodes. The first type are nodes within the dielectric inside the region, which we call *internal nodes* and let \mathbf{V}_i be the vector of potentials on such nodes. The second type are nodes on the equipotential surface of conductors inside the black box with the assumption that all the conductors are ideal conductors. We call them *conductor nodes*, and let \mathbf{V}_c be the vector of potentials on such nodes. Floating conductors are treated as dielectric with $\epsilon = 1000$, although other methods can be used to achieve the same effect. The third type are nodes on the boundary of the black box, which we call *port nodes* as stated before, and let \mathbf{V}_p be the vector of potentials on them. A port node may be on a conductor, but it is still called a port node. In such a case, conductors inside of the black box extends to the outside, \mathbf{V}_c is uniquely determined by \mathbf{V}_p .

In a capacitance extraction problem, in order to compute the flux on a Gaussian

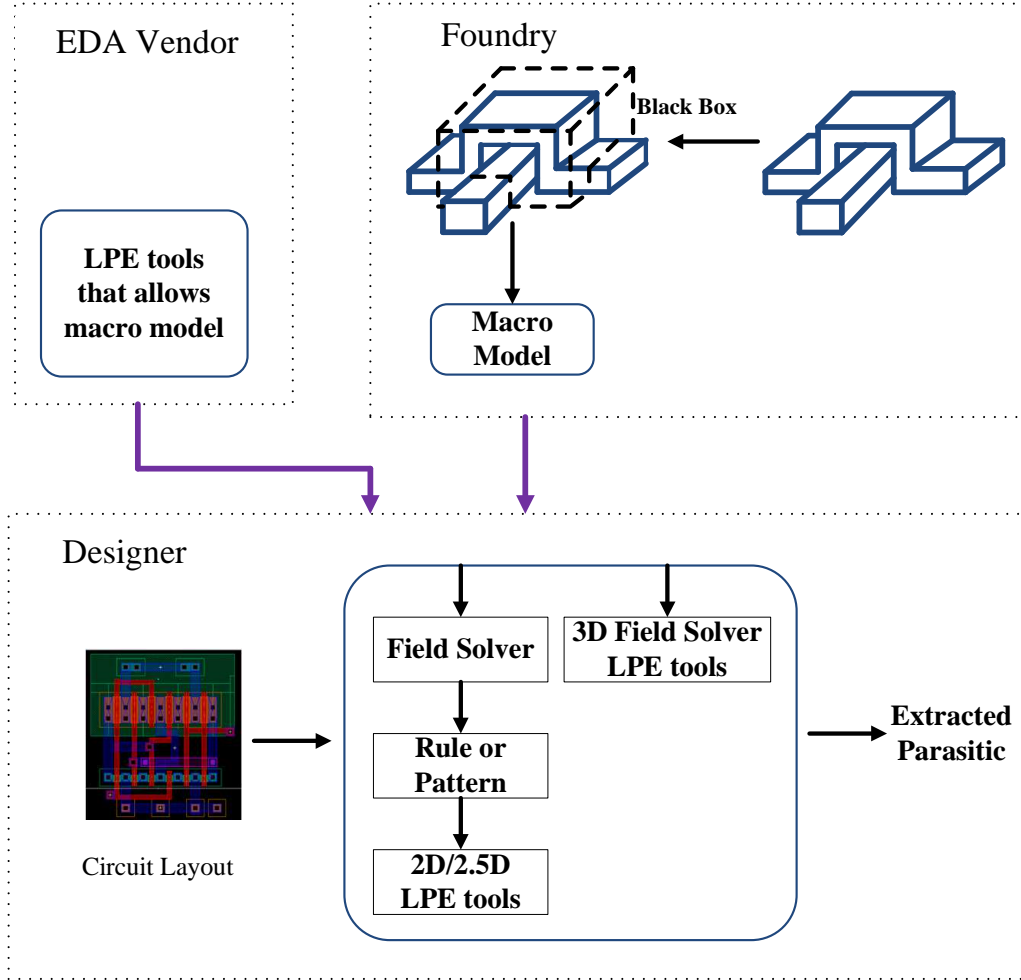


Figure 2.1: Macro model construction and application flow for foundry, EDA vendor and designer.

surface, the potential distribution over the whole domain must be determined. When such dependency relationship within the black box is hidden, the macro model matrix \mathbf{M} enables the interaction of the black box with the neighbors outside without revealing detailed information inside of the black box.

Use FDM [11] on the mesh, the Laplacian equation becomes the following linear system. Matrix \mathbf{A} is a 5-point stencil for 2D and 7-point stencil for 3D, except for

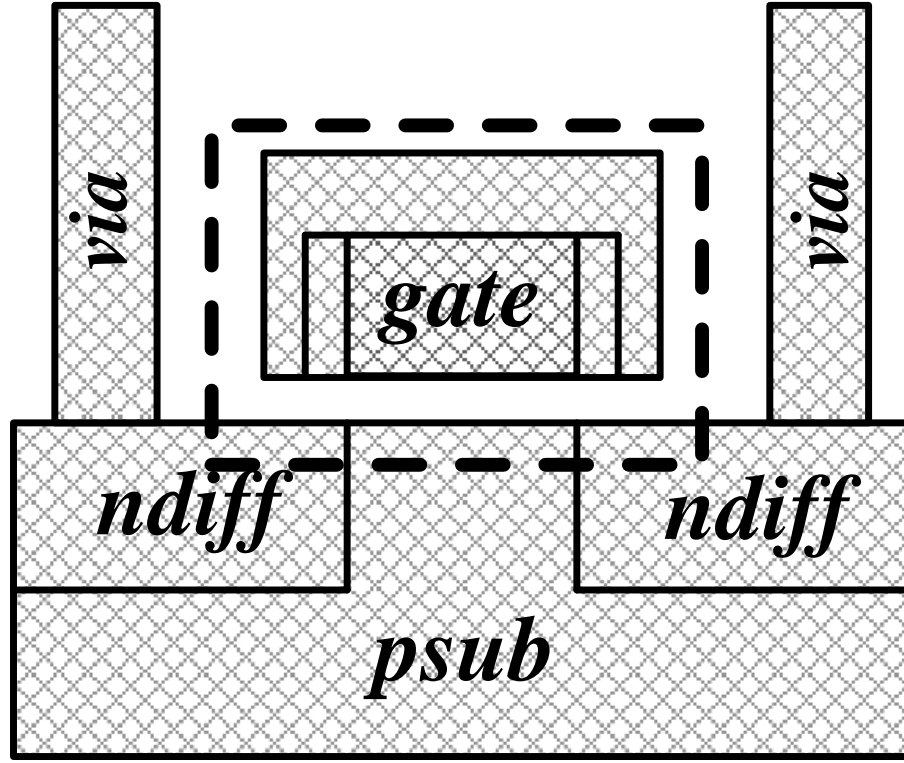


Figure 2.2: Demonstration of 2D MOSFET with black box enclosed by dashed lines.

the rows/columns corresponding to the conductor nodes.

$$\begin{bmatrix} \mathbf{A}_{ii} & \mathbf{A}_{ip} \\ \mathbf{A}_{pi} & \mathbf{A}_{pp} \end{bmatrix} \begin{bmatrix} \mathbf{V}_i \\ \mathbf{V}_p \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{D}_p \end{bmatrix}, \quad (2.1)$$

rearrange the first row of the equation, we get

$$\mathbf{A}_{ii}\mathbf{V}_i = -\mathbf{A}_{ip}\mathbf{V}_p. \quad (2.2)$$

Plugging to the second row of the original system (2.1), we are able to get port

nodes flux

$$\mathbf{D}_p = (\mathbf{A}_{pp} - \mathbf{A}_{pi}\mathbf{A}_{ii}^{-1}\mathbf{A}_{ip})\mathbf{V}_p. \quad (2.3)$$

Since the foundry has complete information on the region, it can compute

$$\mathbf{M} = \mathbf{A}_{pp} - \mathbf{A}_{pi}\mathbf{A}_{ii}^{-1}\mathbf{A}_{ip}, \quad (2.4)$$

and give matrix \mathbf{M} as PDK (Process Design Kit). Matrix \mathbf{M} describes relationship between \mathbf{V}_p and \mathbf{D}_p . The macro model consists of two parts:

- Matrix \mathbf{M} or a low rank approximation of it.
- The coordinates of the port nodes at \mathfrak{B} .

Coordinates of conductor nodes inside of the black box are not needed.

There are far more internal nodes than port nodes. If the number of mesh grid in each direction (X, Y or Z) is k , then the ratio of number of port nodes to the number of internal nodes is $O(k)$ to $O(k^2)$ in 2D, and $O(k^2)$ to $O(k^3)$ in 3D. In such a case, \mathbf{A}_{pp} , \mathbf{A}_{ii} and \mathbf{A}_{ip} , which describe conductors and dielectric distribution inside of the black box, can not be recognized from \mathbf{M} . Therefore it is impossible to derive detailed information about internal nodes from port nodes.

Now we discuss how to efficiently compute \mathbf{M} defined in (2.4). Matrix \mathbf{A}_{ii} is sparse and SPD (symmetric positive definite). If the dimension of \mathbf{A}_{ii} is small, a direct factorization of \mathbf{A}_{ii} will be sufficient to compute \mathbf{M} . If \mathbf{A}_{ii} is large, iterative methods can be used to compute $\mathbf{A}_{pi}\mathbf{A}_{ii}^{-1}\mathbf{A}_{ip}$ as follows. Let

$$\mathbf{X} = \mathbf{A}_{pi}\mathbf{A}_{ii}^{-1}\mathbf{A}_{ip} = \mathbf{A}_{pi}\mathbf{Y}, \quad (2.5)$$

where

$$\mathbf{Y} = \mathbf{A}_{ii}^{-1} \mathbf{A}_{ip} . \quad (2.6)$$

Therefore, we first solve

$$\mathbf{A}_{ii} \mathbf{Y} = \mathbf{A}_{ip} , \quad (2.7)$$

for \mathbf{Y} using iterative method, and then compute $\mathbf{X} = \mathbf{A}_{pi} \mathbf{Y}$. Please note that \mathbf{X} and \mathbf{Y} are matrices, not vectors. Therefore, there are multiple right-hand-sides in linear systems (2.7). A good preconditioner can be constructed, to be used repeatedly for the multiple right-hand-sides.

2.4 Using of Macro Model by FDM field Solver

In this section, we shall discuss how EDA vendors can integrate the macro model into LPE tools.

LPE tools are either based directly on field solvers, or indirectly, through rules and patterns which must be determined using field solvers. Therefore it is sufficient to show how field solvers can utilize the macro model.

Fig. 2.2 is a 2D example application where a macro model for the black box is given, but no dielectric and conductor information in the black box is given. In this example, we are asked to extract coupling capacitance between gate and drain, as well as gate and source.

We first perform discretization by adding mesh grid to form a *global mesh*. If the step size of macro model ports is sufficiently dense, then the global mesh uses the same step size. However if the step size of macro model ports is too coarse, we add more to the global mesh. On the other hand, if the step size of macro model

ports is too dense, we may skip some in the global mesh. If either of these situations happens, interpolation or extrapolation is used to map the macro model ports to the global mesh.

Given the macro model \mathbf{M} , we can build linear system to do capacitance extraction for the entire domain. Using FDM, we have the following linear system for the domain outside of the black box:

$$\begin{bmatrix} \mathbf{B}_{\text{oo}} & \mathbf{B}_{\text{op}} \\ \mathbf{B}_{\text{po}} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{\text{o}} \\ \mathbf{V}_{\text{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{\text{o}} \\ \mathbf{0} \end{bmatrix}, \quad (2.8)$$

where \mathbf{V}_{o} and \mathbf{D}_{o} are the vectors of potential and electric flux on all nodes respectively outside of the black box, \mathbf{V}_{p} is the vector of potential on ports. Matrix \mathbf{B}_{oo} is a Laplacian matrix, describing the interaction between nodes outside of the black box. Matrices \mathbf{B}_{op} and \mathbf{B}_{po} describe the interaction between nodes outside of the black box and port nodes on the black box. Matrices \mathbf{B}_{oo} , \mathbf{B}_{op} and \mathbf{B}_{po} are all sparse matrices.

Since the macro model also contains the coordinates of port nodes, the connection between nodes outside of the black box and port nodes on the black box can be easily established. For those port nodes representing conductors inside of the black box, a corresponding node outside of the black box is created and an equivalence is established by \mathbf{B}_{oo} and \mathbf{B}_{op} .

The flux from outside of the black box cancels out the flux from inside of the black box at port nodes. This explains the dependency between \mathbf{V}_{p} and \mathbf{V}_{o} :

$$\mathbf{V}_{\text{p}} = -\mathbf{M}^{-1}\mathbf{B}_{\text{po}}\mathbf{V}_{\text{o}}. \quad (2.9)$$

Substituting \mathbf{V}_p in (2.8), we have

$$\mathbf{D}_o = (\mathbf{B}_{oo} - \mathbf{B}_{op}\mathbf{M}^{-1}\mathbf{B}_{po})\mathbf{V}_o. \quad (2.10)$$

Matrices \mathbf{M} or its approximation is provided by the foundry. In fact, to simplify computation, the foundry may provide \mathbf{M}^{-1} directly or its approximation.

There are many known techniques to solve the linear system (2.10). Basically, the given potential of conductors are brought into (2.10), resulting in elimination of some rows/columns and modification of the right hand side. Then the linear system can be solved either by a direct factorization or by an iterative method.

When solving the system by an iterative method, one needs to use a preconditioner to accelerate the convergence to the solution. A block diagonal preconditioner consisting of the two diagonal blocks of the system matrix can be used. An alternative is to use a block upper triangular system. The system matrix A and the preconditioner \tilde{A} are shown below:

$$A = \begin{bmatrix} \mathbf{B}_{oo} & \mathbf{B}_{op} \\ \mathbf{B}_{po} & \mathbf{M} \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} \mathbf{B}_{oo} & \mathbf{B}_{op} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}. \quad (2.11)$$

To analyze the effect of the preconditioner, we estimate the eigenvalues λ of the preconditioned matrix that satisfy the equation:

$$\begin{bmatrix} \mathbf{B}_{oo} & \mathbf{B}_{op} \\ \mathbf{B}_{po} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{B}_{oo} & \mathbf{B}_{op} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}. \quad (2.12)$$

This simplifies (2.12) to the condition that either $\lambda = 1$ or λ satisfies:

$$(\lambda - 1)\mathbf{M}\mathbf{y} = -\mathbf{B}_{\text{po}}\mathbf{B}_{\text{oo}}^{-1}\mathbf{B}_{\text{op}}. \quad (2.13)$$

Since the number of eigenvalues that differ from unity equals the rank of \mathbf{M} , the number of iterations required for convergence does not exceed the number of port node variables.

The block preconditioner requires factoring \mathbf{M} and \mathbf{B}_{oo} . Since \mathbf{M} can be factored once and used many times, the challenge is to factor \mathbf{B}_{oo} quickly. An alternative is to replace \mathbf{B}_{oo} with the identity matrix or an approximation derived from incomplete factorization of \mathbf{B}_{oo} such as IC(0). Each case can be analyzed to predict the quality of the resulting preconditioner. One can also compute an incomplete factorization of A directly using the standard algorithm.

2.5 Construction and Using of Macro Model for FEM

Instead of FDM, we can apply macro model concept to Finite Element Method (FEM) based extraction process.

With the weak form of the Laplacian equation

$$-\Delta v = 0 \quad (2.14)$$

defined as

$$(\nabla\psi, \nabla v) = 0, \quad (2.15)$$

assuming ψ is a test function, the basic principle of FEM is to apply finite dimensional approximation to solution v in infinite dimensional space.

In general, the formulation of FEM [12] is based on the discretized elements. In each element e , the discretized local potential V^e is

$$V^e = \sum_{j=0}^N V_j^e \psi_j^e, \quad (2.16)$$

where $\{\psi_j^e\}_{j=1}^N$ is a set of N polynomial basis functions defined on the nodes of the element, and V_j^e is the unknown expansion coefficients for the potential distribution approximation. The Laplacian equation can be discretized into

$$[\mathbf{K}] \cdot \vec{V} = \vec{b}, \quad (2.17)$$

where the global stiffness matrix $[\mathbf{K}]$ is constructed by the assembly process on the local stiffness matrix $[\mathbf{K}^e]$ which is defined as

$$\mathbf{K}_{ij}^e = \int_{\Omega^e} \epsilon^e \nabla \psi_i^e \cdot \nabla \psi_j^e d\Omega^e. \quad (2.18)$$

The macro model for FEM can be built in the foundry side for any sensitive region indicated by the black box. Once the black box is defined, Laplacian equation can be discretized as

$$[\mathbf{K}_B] \begin{bmatrix} \mathbf{V}_i \\ \mathbf{V}_p \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ip} \\ \mathbf{K}_{pi} & \mathbf{K}_{pp} \end{bmatrix} \begin{bmatrix} \mathbf{V}_i \\ \mathbf{V}_p \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_M \mathbf{V}_p \end{bmatrix}, \quad (2.19)$$

where $[\mathbf{K}_B]$ is the stiffness matrix as an assemble of the local stiffness matrix for the black box, \mathbf{V}_i and \mathbf{V}_p are potential vectors for internal nodes and port nodes respectively. Since the port nodes are the only interface with the outside of the black box, we simplify $\mathbf{K}_{pi} \mathbf{V}_i + \mathbf{K}_{pp} \mathbf{V}_p$ in terms of \mathbf{V}_p and define such dependency matrix

\mathbf{K}_M as a macro stiffness matrix. From (2.19), it is easy to get

$$\mathbf{K}_M = \mathbf{K}_{pp} - \mathbf{K}_{pi}\mathbf{K}_{ii}^{-1}\mathbf{K}_{ip}. \quad (2.20)$$

Due to the FEM equation is actually the weak form of the conservation equation and its performance depends on the choice of basis function, the formula of \mathbf{K}_M is not the same as (2.4), though they are similar and close to each other in norm.

As shown in Fig. 2.1, the macro model for FEM provided by the foundries and LPE tools provided by the EDA vendors should be well prepared before the designers perform extraction for a system with black box. For the system outside of the black box, assuming the corresponding stiffness matrix \mathbf{K}' is composed of \mathbf{K}'_{oo} , \mathbf{K}'_{op} , \mathbf{K}'_{po} and \mathbf{K}'_{pp} , we can directly assemble \mathbf{K}_M to the port nodes part as below:

$$\begin{bmatrix} \mathbf{K}'_{oo} & \mathbf{K}'_{op} \\ \mathbf{K}'_{po} & \mathbf{K}'_{pp} + \mathbf{K}_M \end{bmatrix} \begin{bmatrix} \mathbf{V}_o \\ \mathbf{V}_p \end{bmatrix} = \mathbf{0}. \quad (2.21)$$

Once the potential distribution is determined, it is trivial to calculate capacitance.

2.6 Construction and Using of Macro Model for Floating Random Walk Method

Floating Random Walk (FRW) method [13] is composed of a series of Monte Carlo simulation. In order to calculate capacitance of a conductor C , a point i on the Gaussian surface of C is randomly selected as the starting point for each FRW path. Then a transition domain D is created (i.e., a square for 2D and a cube for 3D) which encloses i . It is limited in size by the nearest conductor. A point j on the surface of D is selected according to a probability density function which is defined to be equal to the Green's function [13]. If the point j is on a conductor,

the path is ended. Otherwise, a new transition domain will be created for j and the aforementioned process will be repeated until the path reaches a conductor. Averaging the potential on those conductors for FRW paths starting from i gives the potential for i .

We restrict random walks to start and end at port nodes of the black box without leaving the black box. This process can be simplified as one hop transition from port nodes to port nodes. The macro model over the black box describes this one hop transition as matrix \mathbf{R} of dimension $p \times p$ in Def. 2, where p is the number of port nodes on the black box. $\mathbf{V}_{\mathbf{p}}^+$ is a local potential vector for all port nodes without any contribution outside of the black box, while $\mathbf{V}_{\mathbf{p}}$ is a global potential vector including contribution from both inside and outside of the black box. Each entry r_{ij} of \mathbf{R} represents the probability that a FRW path enters the black box at port node i and exits the black box at port node j . If the FRW path hits a conductor inside of the black box, it will be regarded as exiting at the corresponding port node at the boundary, and the path is terminated. Note that we allow a FRW path entering the black box at node i , either exit at node j ($j \neq i$) with probability r_{ij} , or exit at node i itself with probability r_{ii} .

Definition 2: Macro Model on FRW method in 2D

Black Box Region:

$$[x_{min}, x_{max}] \times [y_{min}, y_{max}]$$

Port Nodes:

$$P_k = (x_k, y_k), \quad k = 1, 2, \dots, p.$$

Markov Transition Matrix:

$$\mathbf{R}_{p \times p} \text{ specifying } \mathbf{V}_{\mathbf{p}}^+ = \mathbf{R}\mathbf{V}_{\mathbf{p}}$$

With the definition of Markov transition matrix \mathbf{R} , a FRW field solver can compute capacitance without ever entering the black box. Whenever a FRW path reaches the boundary of the black box at port node i , it will either go out of the black box, or enter the black box. In 2D at uniform dielectric, if i is on the black box corner, it will enter the black box with probability $1/4$. If i is on the black box edge, this probability will be $1/2$. When it gets into the black box, the FRW path will select a port node or a conductor node j inside of the black box with probability r_{ij} . If it arrives at conductor, this FRW path stops. Otherwise, the path continues at node j .

To compute \mathbf{R} , the following FRW procedure can be used: for each port node i , many FRW paths are initiated without leaving the region. If a FRW path hits a node j , either on the boundary or a conductor inside of the black box, matrix entry (i, j) increases by 1. We normalize each row by dividing its sum when it is convergent.

From previous statement, in the macro model the relationship between ports flux and potential can be summarized as $\mathbf{D}_p = \mathbf{M}\mathbf{V}_p$. Theorem. 1 shows that we can construct \mathbf{R} from \mathbf{M} .

Theorem 1. *Let \mathbf{R} be the Markov transition matrix defined for a black box in the FRW method, \mathbf{M} be the matrix describing port flux dependency in terms of potential, Λ_p be the diagonal matrix extracted from the Laplacian matrix \mathbf{A}_{pp} .*

$$\mathbf{R} = \mathbf{I} - \Lambda_p^{-1}\mathbf{M}. \quad (2.22)$$

Proof. Random walk on the finite grids is restricted to walk to adjacent grid nodes each step, where the grid mesh could be uniform or non-uniform. Since the FRW method is an approximation of the random walk on grids, we adopt the latter for analysis.

Define a one-step transition matrix \mathbf{P} , where p_{ij} is the probability that the random walk goes from grid node i to its neighboring grid node j in one step. For non-adjacent nodes i and j , $p_{ij} = p_{ji} = 0$ since i and j is not accessible from each other in one step walking. For each node and its neighbors j_1, j_2, \dots, j_k , $\sum_{k=1} p_{ij_k} = 1$. This happens to be a row in the Laplacian matrix \mathbf{A} in (2.1) with the following conditions: 1) all the diagonal entries are zeros, since node i is not allowed to walk to itself in one step; 2) sum of each row is 1. Therefore, we have (2.23) where diagonal matrices $-\mathbf{\Lambda}_i$ and $-\mathbf{\Lambda}_p$ will normalize rows corresponding to internal nodes and port nodes respectively assuming all diagonal entries of matrix \mathbf{A} are negative for simplicity. The identity matrix will eliminate the diagonal entries of -1. We define sub-matrices of \mathbf{P} to be $\mathbf{P}_{ii}, \mathbf{P}_{pp}, \mathbf{P}_{ip}$ and \mathbf{P}_{pi} , describing transition probabilities from internal nodes to internal nodes, port nodes to port nodes, internal nodes to port nodes, and port nodes to internal nodes respectively.

$$\begin{aligned}
\mathbf{P} &= \begin{bmatrix} \mathbf{P}_{ii} & \mathbf{P}_{ip} \\ \mathbf{P}_{pi} & \mathbf{P}_{pp} \end{bmatrix} \\
&= \begin{bmatrix} -\mathbf{\Lambda}_i^{-1} & 0 \\ 0 & -\mathbf{\Lambda}_p^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{ii} & \mathbf{A}_{ip} \\ \mathbf{A}_{pi} & \mathbf{A}_{pp} \end{bmatrix} + \mathbf{I} \\
&= \begin{bmatrix} -\mathbf{\Lambda}_i^{-1} \mathbf{A}_{ii} + \mathbf{I} & -\mathbf{\Lambda}_i^{-1} \mathbf{A}_{ip} \\ -\mathbf{\Lambda}_p^{-1} \mathbf{A}_{pi} & -\mathbf{\Lambda}_p^{-1} \mathbf{A}_{pp} + \mathbf{I} \end{bmatrix}.
\end{aligned} \tag{2.23}$$

The random walk procedure for computing \mathbf{R} starts at port nodes, goes through internal nodes, and ends at the port nodes. There are infinite possible paths to walk from one port node to another port node. For example, it can be one step walking from any port node to any other port node with probability \mathbf{P}_{pp} ; it can be two steps walking from any port node to any internal node, and then from the internal node to

any port node, which has probability $\mathbf{P}_{\mathbf{pi}}\mathbf{P}_{\mathbf{ip}}$; it can also be three steps walking with probability $\mathbf{P}_{\mathbf{pi}}\mathbf{P}_{\mathbf{ii}}\mathbf{P}_{\mathbf{ip}}$ and so on. Considering all possible cases, we could express \mathbf{R} in infinite series as,

$$\mathbf{R} = \mathbf{P}_{\mathbf{pp}} + \sum_{k=0}^{\infty} \mathbf{P}_{\mathbf{pi}}(\mathbf{P}_{\mathbf{ii}})^k \mathbf{P}_{\mathbf{ip}}. \quad (2.24)$$

We prove $\|\mathbf{P}_{\mathbf{ii}}\|_2 < 1$ first. Since all entries of $\mathbf{P}_{\mathbf{ii}}$ are non-negative and sum of each row or column is less than or equal to 1,

$$\|\mathbf{P}_{\mathbf{ii}}\|_1 = \|\mathbf{P}_{\mathbf{ii}}\|_{\infty} \leq 1. \quad (2.25)$$

As a special case of Hölder's inequality [14],

$$\|\mathbf{P}_{\mathbf{ii}}\|_2 \leq \sqrt{\|\mathbf{P}_{\mathbf{ii}}\|_1 \|\mathbf{P}_{\mathbf{ii}}\|_{\infty}} \leq 1. \quad (2.26)$$

However, if $\|\mathbf{P}_{\mathbf{ii}}\|_2 = 1$, then 1 is the maximum eigenvalue of $\mathbf{P}_{\mathbf{ii}}$ based on the fact that $\mathbf{P}_{\mathbf{ii}}$ is symmetric,

$$\det(\mathbf{I} - \mathbf{P}_{\mathbf{ii}}) = \det(\Lambda_{\mathbf{i}}^{-1} \mathbf{A}_{\mathbf{ii}}) = 0, \quad (2.27)$$

which contradicts with the fact that $\Lambda_{\mathbf{i}}^{-1} \mathbf{A}_{\mathbf{ii}}$ is symmetric positive definite and $\det(\Lambda_{\mathbf{i}}^{-1} \mathbf{A}_{\mathbf{ii}}) > 0$. Therefore, $\|\mathbf{P}_{\mathbf{ii}}\|_2 < 1$.

From Theorem on Neumann Series [15], we know that $\mathbf{I} - \mathbf{P}_{\mathbf{ii}}$ is invertible, and

$$\sum_{k=0}^{\infty} (\mathbf{P}_{\mathbf{ii}})^k = (\mathbf{I} - \mathbf{P}_{\mathbf{ii}})^{-1}. \quad (2.28)$$

Then

$$\begin{aligned}
\mathbf{R} &= \mathbf{P}_{\text{pp}} + \mathbf{P}_{\text{pi}}(\mathbf{I} - \mathbf{P}_{\text{ii}})^{-1}\mathbf{P}_{\text{ip}} \\
&= -\Lambda_{\text{p}}^{-1}\mathbf{A}_{\text{pp}} + \mathbf{I} - \Lambda_{\text{p}}^{-1}\mathbf{A}_{\text{pi}}\mathbf{A}_{\text{ii}}^{-1}\Lambda_{\text{i}}(-\Lambda_{\text{i}}^{-1}\mathbf{A}_{\text{ip}}) \\
&= \mathbf{I} + \Lambda_{\text{p}}^{-1}(\mathbf{A}_{\text{pi}}\mathbf{A}_{\text{ii}}^{-1}\mathbf{A}_{\text{ip}} - \mathbf{A}_{\text{pp}}) \\
&= \mathbf{I} - \Lambda_{\text{p}}^{-1}\mathbf{M}.
\end{aligned} \tag{2.29}$$

□

2.7 Profile Simplification

Sometimes, high accuracy extraction is not required, or the foundries want to provide simplified profile with sensitive information distorted to the users. In such a case, macro model is not needed. However, the macro model can be used to construct simplified profile for traditional LPE tools. Then, designers will combine the simplified profile provided by the foundries and LPE tools provided by the EDA vendors to perform extractions.

2.7.1 Concept of Equivalent Profile

We define that two profiles are equivalent if their electromagnetic behavior are equal at an enclosing Gaussian surface. To be precise, let \mathbb{B}_1 and \mathbb{B}_2 be two black boxes whose surface are both \mathfrak{B} , then we say \mathbb{B}_1 and \mathbb{B}_2 are equivalent if

$$\frac{\|\mathbf{M}_1 - \mathbf{M}_2\|_{\infty}}{\|\mathbf{M}_1\|_{\infty}} < \epsilon, \tag{2.30}$$

where \mathbf{M}_1 and \mathbf{M}_2 are macro model matrices, ϵ is criteria which should be small enough.

The flux difference at \mathfrak{B} is maximum when potential at all port nodes is 1, and

flux contribution from each port node is in the same direction, either increase or decrease. This explains why infinity norm is used to measure whether two black boxes behave the same.

From classical electromagnetic theory, it is easy to derive that the profiles \mathbf{P}_1 and \mathbf{P}_2 are equivalent if there exists equivalent \mathbb{B}_1 and \mathbb{B}_2 satisfying \mathbb{B}_1 encloses \mathbf{P}_1 , and \mathbb{B}_2 encloses \mathbf{P}_2 .

Then extraction based on equivalent profile will generate the same result. If a section of profile is complex yet appears at many places in a circuit, we can use a simplified equivalent profile to reduce the extraction time.

2.7.2 Finding Equivalent Profile

In the domain determined by original profile, we define a black box region randomly, keep its surface port nodes intact. In order to find equivalent profile, we can make any changes within the black box, through either conductor shapes modification or different dielectric distribution. There is no specific restriction on how to make the updated profile equivalent with the old one. However, some heuristic ideas can be adopted to correct those changes. For example, when we step forward to extend boundary of a conductor and make it close to the surface, it will result in great flux changes on the corresponding port nodes. Then we need to step backward to mitigate this effect, through adjusting dielectric around those port nodes. Through such repetitive stepwise process, we step forward to make new geometric or parameter modification, and step backward to reduce great changes on port nodes, until new profile satisfies criteria as shown in Fig. 2.3. Such heuristic approach works fine in most cases, though it is not guaranteed that the equivalent profile is optimal and unique.

A simplified profile not only reduces computational cost, but also keeps sensitive

Table 2.1: 2D MOSFET: Comparison of extraction without black box and with different black boxes.

disc. step h	black box	capacitance						run time	
		gate		drain-source				macro model (s)	total system (s)
		total (fF)	err (%)	in box (fF)	out box (fF)	total (fF)	error (%)		
1/10	No	0.261	-	-	-	0.248	-	-	9
	Yes	0.261	0.00	0.153	0.095	0.248	0.00	1	5
1/20	No	0.258	-	-	-	0.246	-	-	32
	Yes	0.258	0.00	0.152	0.094	0.246	0.00	11	45

information secret.

2.8 Experimental Results

The algorithms are implemented in C++ and tested with industry cases. The focus of the simulation is on i) the accuracy of capacitance extraction using the macro model v.s. without using the macro model, ii) robustness for different mesh grids and iii) run time to construct the macro model.

2.8.1 Correctness

For the 2D case, we use traditional FDM to extract coupling capacitance of gate and source, drain first. A black box is added in Fig. 2.2. The dependency matrix \mathbf{M} is pre-calculated. It is integrated into global FDM matrix assuming the black box is unknown. The final extraction results are exactly the same as those from the traditional approach shown in Table 2.1.

Simulations on 3D FinFET model are demonstrated in Fig. 2.4. The basic structure and two arbitrarily chosen black boxes are shown in Fig. 2.5 and Fig. 2.6 respectively. Similarly, the computation results in Table 2.2 do not result in any accuracy loss after applying different black boxes. It shows regardless of how to

Table 2.2: 3D FinFET: Comparison of extraction without black box and with different black boxes.

disc. step h	black box	capacitance						run time	
		gate		drain-source				macro model (s)	total system (s)
		total (fF)	err (%)	in box (fF)	out box (fF)	total (fF)	error (%)		
1/4	No	0.429	-	-	-	0.363	-	-	14
	Fig.2.5 Box	0.429	0.00	0.219	0.144	0.363	0.00	1	15
	Fig.2.6 Box	0.429	0.00	0.056	0.307	0.363	0.00	1	16
1/8	No	0.412	-	-	-	0.347	-	-	342
	Fig.2.5 Box	0.412	0.00	0.208	0.139	0.347	0.00	3	408
	Fig.2.6 Box	0.412	0.00	0.056	0.291	0.347	0.00	14	449

define the black box, the capacitance are the same.

2.8.2 Robustness

In order to verify the robustness of macro model, we choose different discretization steps for the macro model mesh and global mesh. Interpolation and extrapolation are used to non-matching port nodes along different discretization boundary. When the discretization steps for macro and global are different by 2X, there are about 1% errors for 2D case shown in Table. 2.3 and 3D FinFet in Table. 2.4. Macro models with low accuracy can be used where high accuracy is not required, resulting in further gains in efficiency. In practical, the foundries will choose the black box size and step size of the macro model depending on accuracy and data volume of the actual problem, as well as the customer experience.

2.8.3 Run Time

The running time of constructing macro model is not a crucial consideration, since it is conducted by the foundry as a pre-process and can be reusable. However, experiment results still show that the macro model can be efficiently constructed.

Table 2.3: 2D MOSFET: Comparison of extraction with different discretization steps inside and outside of black box.

disc. step h		capacitance					
		gate		drain-source			
global	macro	total (fF)	err (%)	in box (fF)	out box (fF)	total (fF)	error (%)
1/20	1/10	0.261	0.874	0.153	0.095	0.248	0.819
	1/20	0.258	-	0.152	0.094	0.246	-
	1/40	0.258	-0.305	0.151	0.094	0.245	-0.301
1/40	1/20	0.258	0.414	0.152	0.094	0.246	0.386
	1/40	0.257	-	0.151	0.094	0.245	-
	1/80	0.257	-0.146	0.151	0.093	0.244	-0.143

From Table 2.1 and Table 2.2, the time used to construct macro model is less than 25% of the total simulation time depending on the geometrical complexity inside of the black box. Therefore, it is not a bottleneck for the foundry. Besides, in our experiments, the black box has size up to 120 nm×120 nm and 48 nm×48 nm×48 nm for 2D and 3D cases respectively which is sufficiently large, while the grid size is 1 nm which is sufficiently small. Thus, the macro model approach will be feasible for practical cases in industry.

2.8.4 Equivalent Profile Experiment

We implemented two basic profile modifications to verify the accuracy of equivalent profile.

The first experiment is applied on a simple 2D MOSFET with adjacent contacts as depicted in Fig. 2.7. The original MOSFET has raised drain and source which the foundry wants to hide. The experiment result in Table 2.5 shows the capacitance extraction error is less than 1% when the criteria is 0.1%.

Table 2.4: 3D FinFET: Comparison of extraction with different discretization steps inside and outside of black box in Fig. 2.6.

disc. step h		capacitance					
global	macro	gate		drain-source			
		total (fF)	error (%)	in box (fF)	out box (fF)	total (fF)	error (%)
1/4	1/2	0.430	0.342	0.056	0.309	0.365	0.543
	1/4	0.429	-	0.056	0.307	0.363	-
	1/8	0.429	-0.057	0.057	0.306	0.363	-0.165
1/8	1/4	0.413	0.140	0.056	0.292	0.348	0.237
	1/8	0.412	-	0.056	0.291	0.347	-
	1/16	0.412	-0.019	0.056	0.291	0.347	-0.062

Table 2.5: Comparison of equivalent profile for MOSFET gate part with respect to adjacent contacts with $\epsilon = 0.1\%$.

profile	total gate cap (fF)	err (%)	gate-src/drn cap (fF)	err (%)	gate-contact cap (fF)	err (%)
a	0.213	-	0.116	-	0.094	-
b	0.214	0.51	0.117	0.73	0.094	0.40

Table 2.6: Comparison of equivalent profile for trapezoidal contact with $\epsilon = 0.1\%$.

profile	contact cap (fF)	err (%)	adjacent gate cap (fF)	err (%)
a	0.072	-	0.093	-
b	0.073	0.31	0.093	-0.32

The second experiment is to simplify trapezoidal contacts with boundary as shown in Fig. 2.8. We use uniform staircase to approximate such boundary originally. However, it results in too many stair steps, which increases the computational cost. We adjust distance of contact and gate in sensitive region and balance modification in corresponding port node flux. The results in Table 2.6 show equivalent profile provides almost the same coupling relationship with components outside of the black box even though great geometric modification has been made inside of the black box.

2.9 Conclusion

We propose a macro model and equivalent profile to address an increasingly important challenge in parasitic extraction. Macro model can provide accurate result for consistent discretizations in black box and outside, while less than 1% error for different discretizations when linear transformations are required along the black box boundary.

The macro model and equivalent profile will benefit the foundry, the EDA vendors, and circuit designers who are the end users of parasitic extractions. They can reveal enough profile and layout information of advanced process and devices so that parasitic extraction can be performed accurately, yet no sensitive information about the process is compromised. They can also be applied to speed up the field solver computation, which will come free once the macro model feature is included.

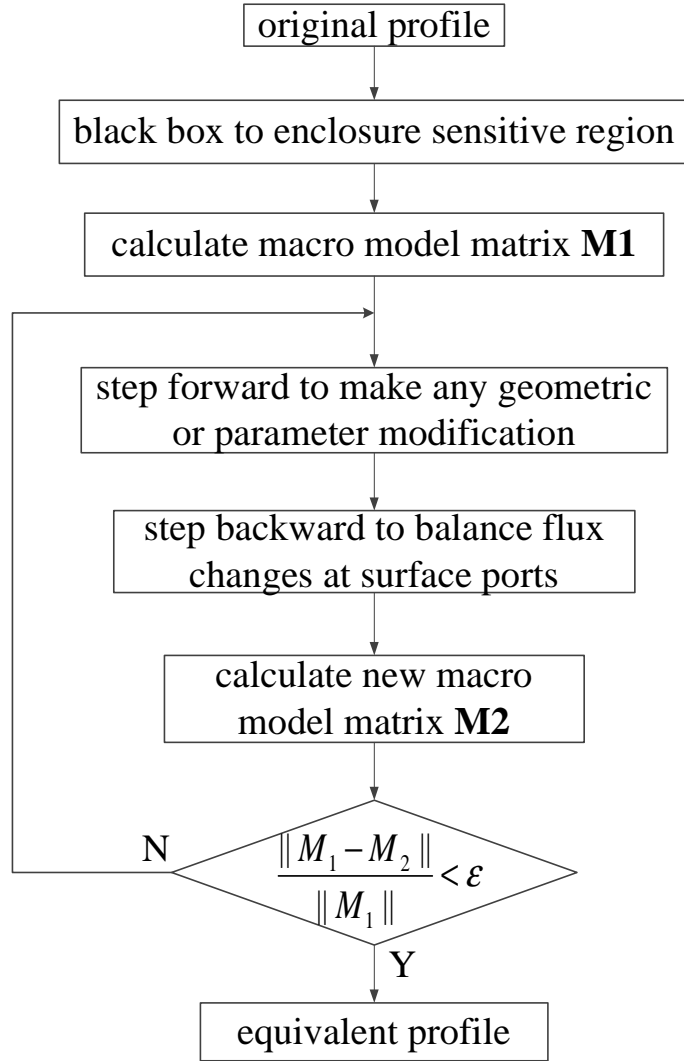


Figure 2.3: Flow diagram of finding equivalent profile.

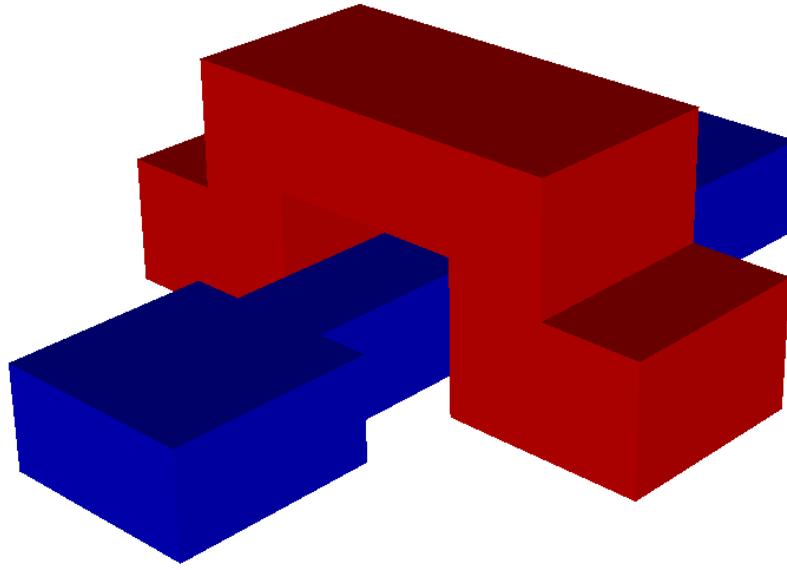


Figure 2.4: 3D FinFET Model.

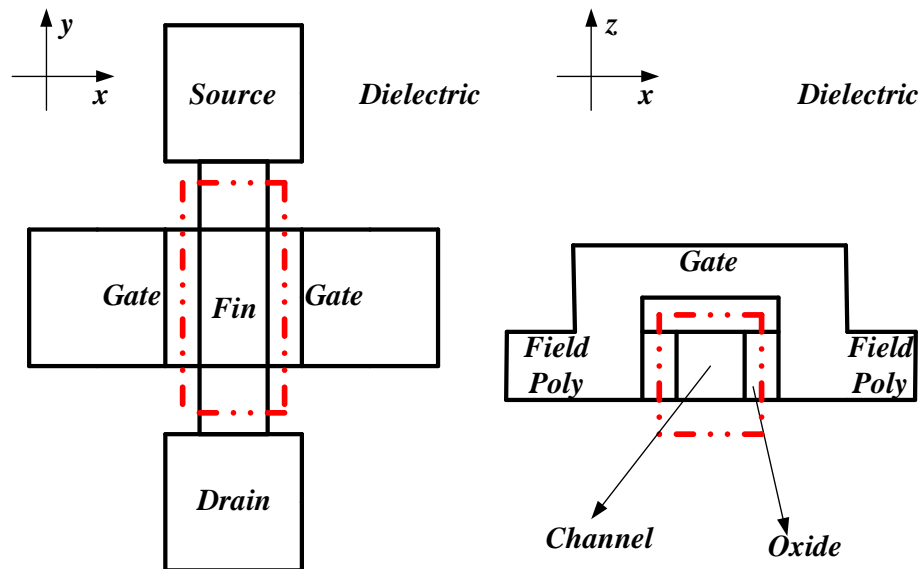


Figure 2.5: Top view and side view of the 3D FinFET structure and the black box around Fin part.

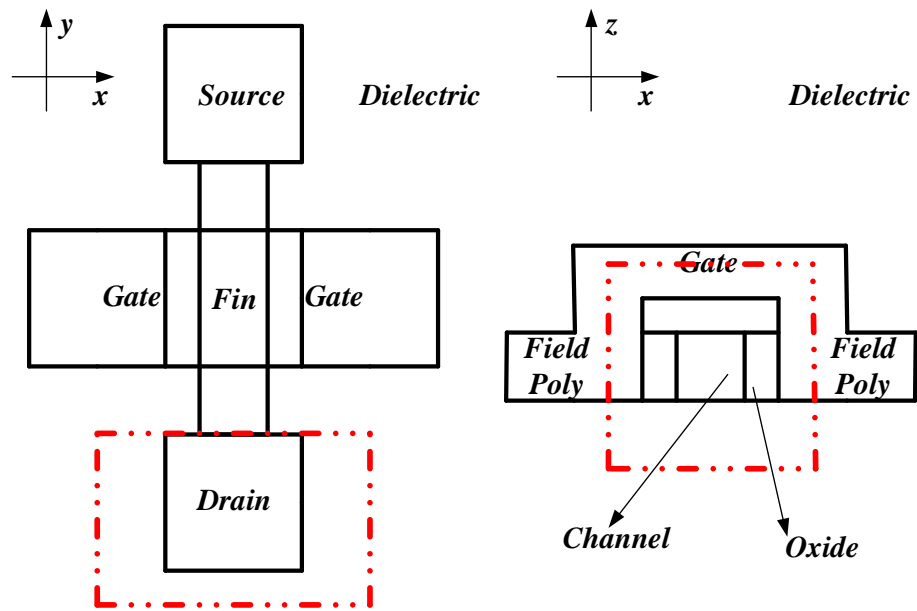
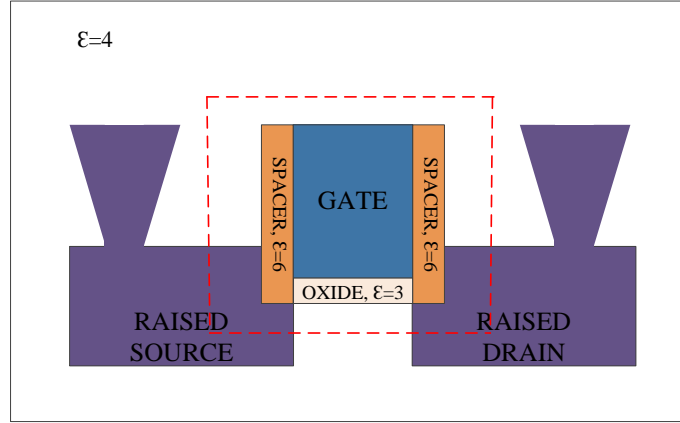


Figure 2.6: Top view and side view of the 3D FinFET structure and the black box around Drain part.



(a)

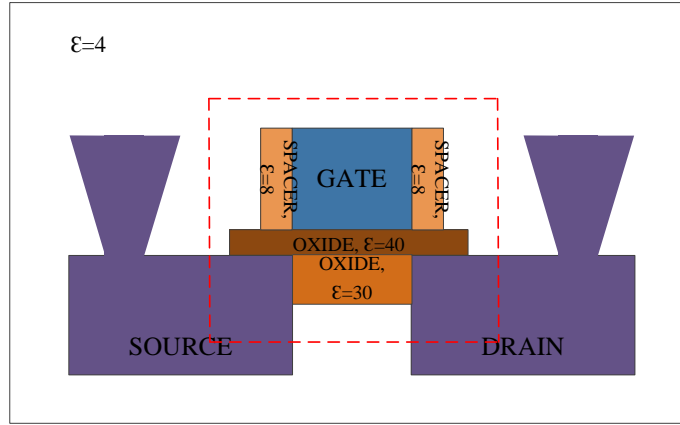


Figure 2.7: Original profile with raised source and drain is shown in (a). Simplified profile is shown in (b). Both profiles produce the same extracted capacitance between device pins and contact.

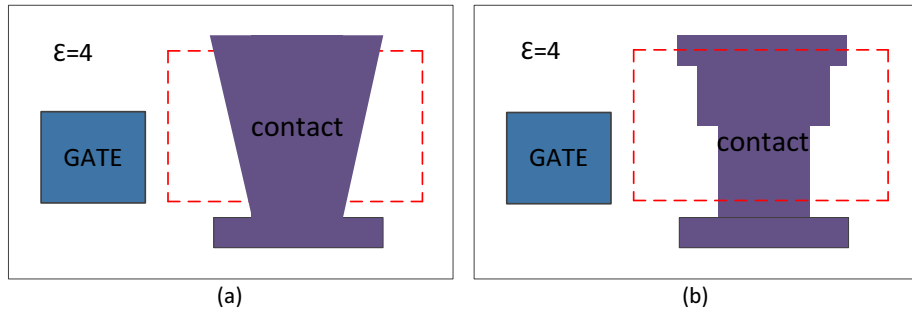


Figure 2.8: The original profile of trapezoidal contact is shown in (a), simplified equivalent profile is shown in (b).

3. CAPACITANCE EXTRACTION WITH PROVABLY GOOD ABSORBING BOUNDARY CONDITIONS

In this chapter, we define a class of absorbing boundary conditions for static PDE problem and approximate them numerically. The performance is evaluated through multiple experiments.

3.1 Background

Parasitic extraction using 3D field solvers has been a recent trend in industry. Implementation of boundary conditions at the outer periphery of the numerical region is an important consideration when performing capacitance extraction with field solvers based on such methods as Finite Difference Method (FDM), Finite Element Method (FEM) or Floating Random Walk (FRW) [13]. Traditional boundary conditions used in FDM/FEM include the Neumann boundary [16] (also called the Perfect Magnetic Conductor or intuitively, reflective wall), and Dirichlet boundary (also called the Perfect Electric Conductor, or intuitively, ground wall, when the potential is set at 0). In FRW, the boundary is implicit. When a random walk hits the enclosing boundary, it is either reflected back, which is equivalent to Neumann boundary, or the walk is omitted from the calculation, resulting in significant error in the field at that point.

It is well known that use of the Neumann boundary to reduce the size of numerical region tends to produce an underestimated capacitance, while using the Dirichlet boundary $V = 0$ tends to produce an overestimated capacitance. Thus to avoid these errors, a very large numerical region is required. This is the case for bulk MOSFET where the ground plane is nearby. For Silicon On Insulator (SOI) where the ground plane is far away from the interconnects, even greater numerical regions

are needed, resulting in much longer run time. This has been a persistent challenge for the industry.

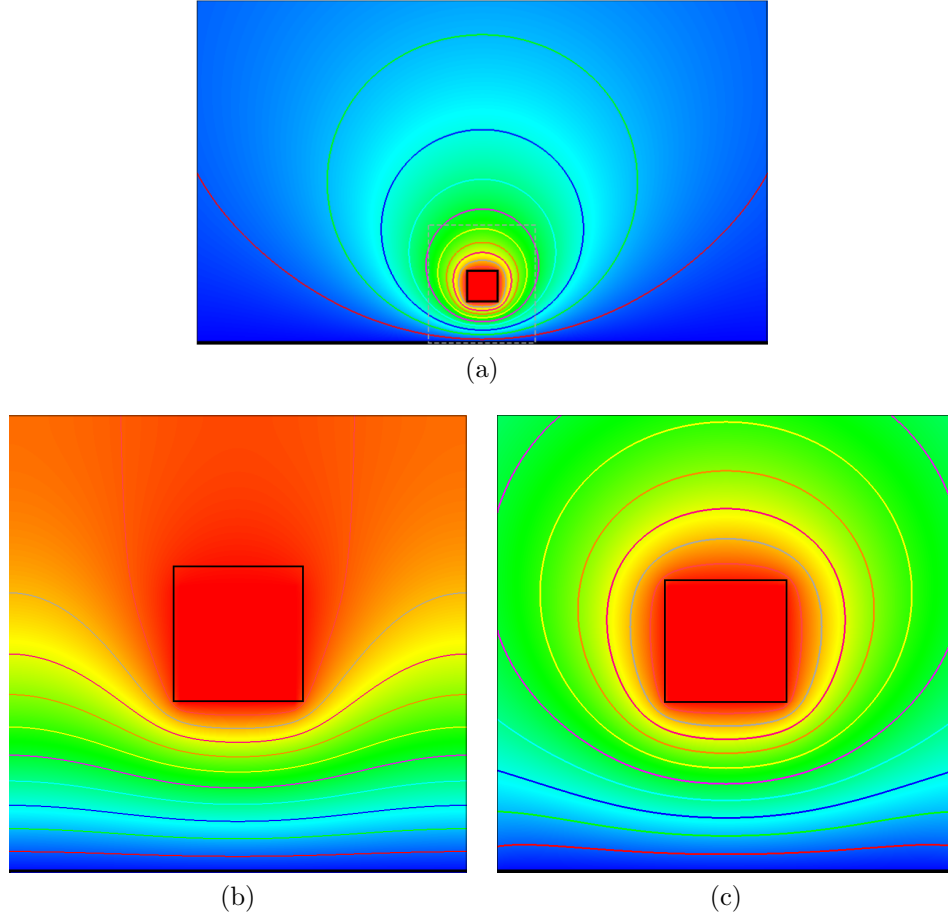


Figure 3.1: Electric potential distribution due to 1 volt conductor above a 0 volt ground plane. The correct distribution computed with a large numerical region is shown in (a). When a small region is used, the traditional Neumann boundary gives inaccurate results as shown in (b), while our absorbing boundary condition gives accurate results as shown in (c).

The Neumann boundary condition requires $\partial V / \partial n = 0$ on the outer boundary, where V is the potential and n is the normal to the boundary. The result of this boundary condition is that a portion of the electric field effectively reflects back into

the numerical region instead of passing smoothly into the boundary in a direction it would have taken if the boundary were not there. The Neumann condition forces the electric to be zero at the boundary, thereby forcing the potential to have zero slope at the boundary as shown in Fig. 3.1(b). Consequently, truncation error is introduced because the electric field and the energy outside the numerical region have been ignored. As an example, the capacitance of the rectangular conductor in Fig. 3.1(b) has an error of 35.6% compared with the exact solution case shown in Fig. 3.1(a). Rather than artificially forcing the electric field to zero along the outer boundary, we introduce a boundary condition that behaves as if the electric field exists far outside of the outer boundary. Fig. 3.1(c) is an implementation of this concept and an illustration of the benefit of having an accurate absorbing boundary condition. The electric potential distribution is very close to the distribution in infinite large domain, thus the error in the conductor capacitance drops to around 1%.

Mathematically inspired absorbing boundary conditions have long been an item of interest in high frequency electromagnetics [17]. These were later followed in [18] by the idea of placing perfectly matched layers (PMLs) on the boundary. In 1993, the absorbing boundary condition was applied to 2D electrostatic problems [19]. Recently, an equivalent dielectric layer method was proposed to reduce the size of the numerical region for 3D devices [20] with the assumption that only the dipolar term of the harmonics is kept. Using a rather complicated approach, known as energy conservation, they derived equations for the artificial layer [20]. However, their equation (7), (8), (9) and (10) are incorrect. In this paper, we derive the correct equation for artificial layers using a much simpler method.

In this chapter, we propose a class of new absorbing boundary conditions, implemented in Cartesian coordinates with detailed error estimates. We present two types

of approximation schemes. The first approach is in concept similar to [21], which is based on a series expansion of the terms in Laplace's equation, resulting in a boundary condition that can be made as accurate as the number of terms that one wishes to add to the series. The second scheme can be considered as a sponge layer [22] around the outer numerical boundary, designed in such a way as to absorb energy and prevent reflections. We provide methods for applying these absorbing boundary techniques to the FEM and FRW method. We validate our method through experiments on interconnects of bulk CMOS and SOI. Both uniform and non-uniform mesh grids are investigated and are shown to achieve similar accuracy with the proposed boundary conditions. It is also shown that the proposed absorbing boundary conditions require fewer mesh nodes than the Neumann boundary condition, and have three to ten times faster run times.

The extraction problem studied in this chapter does not separate intrinsic parasitic, which is also called device parasitic and is already included in SPICE model, vs extrinsic parasitic, which is also called interconnect parasitic. The separation algorithm is out of the scope of this paper, and can be found in [23].

3.2 Absorbing Boundary Conditions

In the capacitance extraction problem, a bounded numerical region encloses a finite number of conductors which reside on the grounded dielectric substrate.

In a charge free region, any point with potential V should satisfy Laplace's equation $\nabla^2 V = 0$, with Dirichlet boundary conditions Γ_D at the conductor surface. Using spherical coordinate (r, θ, ϕ) for any point with potential V , Laplace's equation can be written as

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial V}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial V}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 V}{\partial \phi^2} = 0, \quad (3.1)$$

where r is the radial distance, θ the polar angle, ϕ the azimuthal angle of the point, respectively.

For any point with polar angle and azimuthal angle (θ, ϕ) , the spherical harmonic function of degree l and order m is defined in terms of the associated Legendre polynomials $P_l^m(\cos\theta)$ [24],

$$Y_l^m(\theta, \phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) e^{im\phi}.$$

The general solution of Eqn (3.1) can therefore be expressed as [25]

$$V(r, \theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l (A_{lm} r^l + B_{lm} r^{-l-1}) Y_l^m(\theta, \phi), \quad (3.2)$$

where coefficients A_{lm} and B_{lm} , known as the moments of expansion [26], are determined by the boundary condition Γ_D . For infinite numerical region, it is well known that A_{lm} must be zero. Therefore Eqn (3.2) is simplified to

$$V(r, \theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l B_{lm} r^{-l-1} Y_l^m(\theta, \phi). \quad (3.3)$$

Although Fast Multipole Expansion (FMM) theorem [26] has a similar form, FMM is based on charge, our method is based on Dirichlet boundary condition. Reorganize Eqn (3.3) into a function of r , we have

$$V(r, \theta, \phi) = \frac{C_1(\theta, \phi)}{r} + \frac{C_2(\theta, \phi)}{r^2} + \frac{C_3(\theta, \phi)}{r^3} + \dots, \quad (3.4)$$

where $C_i(\theta, \phi) = \sum_{m=-i}^i B_{im} Y_i^m(\theta, \phi)$ is the coefficient of r^{-i} , which will be simply

labeled as C_i in the rest of the chapter. The electric field is

$$\frac{\partial V}{\partial r} = -\frac{C_1}{r^2} - \frac{2C_2}{r^3} - \frac{3C_3}{r^4} - \dots . \quad (3.5)$$

Under Neumann boundary condition, $\partial V/\partial r = 0$ is set at the outer boundary nodes, which results in an error of $O(1/r^2)$ for approximating the field at the boundary. A naive attempt to improve the Neumann boundary condition would be to set the boundary condition $\partial V/\partial r = -C_1/r^2$. Unfortunately, C_1 is not given. Instead, we define the 1st order Absorbing Boundary Condition (ABC1) at the boundary nodes to satisfy the following condition:

$$\frac{\partial V}{\partial r} + \frac{V}{r} = 0. \quad (3.6)$$

It is easy to verify that under ABC1,

$$\frac{\partial V}{\partial r} = -\frac{V}{r} = -\frac{C_1}{r^2} - \frac{C_2}{r^3} - \frac{C_3}{r^4} - \dots$$

is held at the boundary nodes. Compared with Eqn (3.5), the approximating error of ABC1 for the field at the boundary is $O(1/r^3)$.

To define absorbing boundary condition of an arbitrary order, we introduce the first-order operator \mathcal{A}_1 as

$$\mathcal{A}_1 V = \left(\frac{\partial}{\partial r} + \frac{1}{r} \right) V.$$

Then ABC1 is the corresponding partial differential equation $\mathcal{A}_1 V = 0$.

By observing that

$$-\frac{r}{4}\left(\frac{\partial^2 V}{\partial r^2} + \frac{2V}{r^2}\right) = -\frac{C_1}{r^2} - \frac{2C_2}{r^3} - \frac{7C_3}{2r^4} - \frac{11C_4}{2r^5} + \dots,$$

we can define the second-order operator

$$\mathcal{A}_2 V = \left(\frac{\partial}{\partial r} + \frac{3}{r}\right) \mathcal{A}_1 V,$$

and the corresponding ABC2

$$\mathcal{A}_2 V = \frac{\partial^2 V}{\partial r^2} + \frac{4\partial V}{r\partial r} + \frac{2V}{r^2} = 0. \quad (3.7)$$

The approximating error of ABC2 for the field is $O(1/r^4)$ relative to Eqn (3.5), since

$$\frac{\partial V}{\partial r} = -\frac{r}{4}\left(\frac{\partial^2 V}{\partial r^2} + \frac{2V}{r^2}\right).$$

In general, the n -th order operator can be defined as

$$\mathcal{A}_n V = \prod_{i=1}^n \left(\frac{\partial}{\partial r} + \frac{2i-1}{r}\right) V,$$

and the corresponding n -th order absorbing boundary condition is $\mathcal{A}_n V = 0$, with the error for approximating the field at the boundary nodes being $O(1/r^{n+2})$. This leads to our main result.

Theorem 2. *Let r be the radial distance in spherical coordinate system, then the relative error of electric field for Neumann boundary condition is $O(1/r^2)$, while the errors for ABC1, ABC2 and ABC $_n$ are $O(1/r^3)$, $O(1/r^4)$ and $O(1/r^{n+2})$, respectively. Thus, absorbing boundary conditions are asymptotically more accurate.*

For simplicity, the above discussion assumes uniform dielectric. For a multi-layer dielectric, the potential still decays with high order terms dropped and low order terms $1/r$, $1/r^2$ retained, although the spatially varying dielectrics will increase or decrease the decay rate of the potential near the dielectric interface. Thus, ABCs are still suitable for use in the multi-layer dielectric case. We only implement ABC1 and ABC2 in this paper but for greater accuracy higher order terms can be included at the expense of computation time.

3.3 Implementation in Cartesian Coordinates

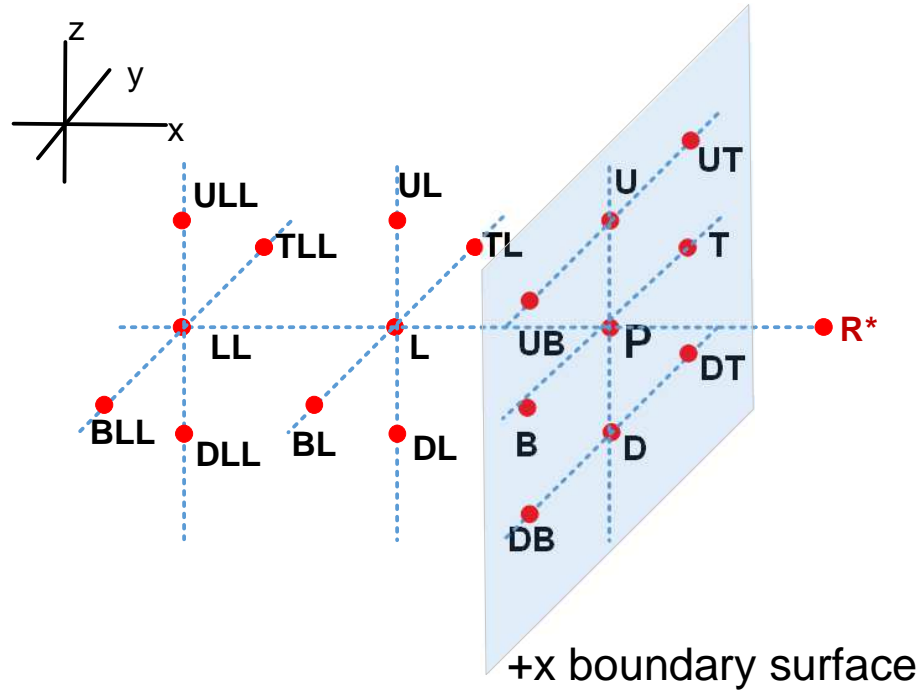


Figure 3.2: The naming of neighbors for node P , which is at the $+X$ boundary. Fictitious node R^* is the mirror of node L outside of the $+X$ boundary, and $V(R^*)$ is approximated by ABC, used to compute $V(P)$.

3.3.1 Absorbing Boundary Condition (ABC1)

Fig. 3.2 shows labels for the P node and its neighboring grid points involved in calculation of the P node potential on the $+X$ face of the numerical region. With the Neumann boundary condition, $V(R^*) = V(L)$ which is not accurate. However, the proposed absorbing boundary condition can provide a better approximation of $V(R^*)$, which in turn improves the accuracy of $V(P)$ when the conservation law is applied.

By the chain rule, we obtain

$$\frac{\partial V}{\partial r} = \frac{\partial V}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial V}{\partial y} \frac{\partial y}{\partial r} + \frac{\partial V}{\partial z} \frac{\partial z}{\partial r} = \frac{\partial V}{\partial x} \frac{x}{r} + \frac{\partial V}{\partial y} \frac{y}{r} + \frac{\partial V}{\partial z} \frac{z}{r}.$$

Eqn (3.6) can be converted to Cartesian coordinates (x, y, z) .

$$\frac{\partial V}{\partial x} = -\frac{1}{x} \left(V + y \frac{\partial V}{\partial y} + z \frac{\partial V}{\partial z} \right). \quad (3.8)$$

Since R^* is the mirror node of L in terms P , and the distance between P and L is D_L , we use the finite difference approximation,

$$\frac{\partial V(P)}{\partial x} = \frac{V(R^*) - V(L)}{2D_L}. \quad (3.9)$$

By replacing the left hand side of Eqn (3.8) with Eqn (3.9) and rearrange the terms, we obtain

$$V(R^*) = V(L) - \frac{2D_L}{x(P)} \left[V(P) + y(P) \frac{\partial V(P)}{\partial y} + z(P) \frac{\partial V(P)}{\partial z} \right]. \quad (3.10)$$

In order to approximate $\partial V(P)/\partial y$, the interpolation polynomial in the Lagrange form [27] is applied in Y -direction. With known potential $V(P)$, $V(T)$, $V(B)$ and y

coordinates $y(P)$, $y(T)$, $y(B)$, the potential $V(y)$ between the node B and T can be written in Lagrange interpolation form

$$\begin{aligned} V(y) = & V(P) \frac{(y - y(B))(y - y(T))}{(y(P) - y(B))(y(P) - y(T))} \\ & + V(T) \frac{(y - y(B))(y - y(P))}{(y(T) - y(B))(y(T) - y(P))} \\ & + V(B) \frac{(y - y(P))(y - y(T))}{(y(B) - y(P))(y(B) - y(T))}. \end{aligned}$$

In general, we define the coefficient of Lagrange interpolating polynomial in the vector form

$$\begin{aligned} \mathcal{L}(x|_{x_1, x_2, x_3}) &\triangleq [\mathcal{L}_1(x), \mathcal{L}_2(x), \mathcal{L}_3(x)]|_{x_1, x_2, x_3} \\ &= \left[\frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}, \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}, \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} \right]. \end{aligned}$$

Then, the partial derivative $\partial V(P)/\partial y$ can be written as the linear combination of $V(P)$, $V(T)$ and $V(B)$. The coefficients depend on the derivative of function \mathcal{L} ,

$$\frac{\partial V(P)}{\partial y} = \mathcal{L}'(y|_{y(P), y(T), y(B)}) \Big|_{y(P)} \cdot [V(P), V(T), V(B)]^T.$$

Similarly, the partial derivative $\partial V(P)/\partial z$ can be approximated through

$$\frac{\partial V(P)}{\partial z} = \mathcal{L}'(z|_{z(P), z(U), z(D)}) \Big|_{z(P)} \cdot [V(P), V(U), V(D)]^T.$$

By substituting the approximation equations above into Eqn (3.10), we can express

the potential at R^* in terms potential of neighboring nodes

$$\begin{aligned} V(R^*) &= C_L V(L) + C_T V(T) + C_B V(B) + C_U V(U) \\ &\quad + C_D V(D) + C_P V(P), \end{aligned} \tag{3.11}$$

where

$$\begin{aligned} C_L &= 1, \\ C_T &= -\frac{2D_L}{x(P)} y(P) \mathcal{L}'_2(y|_{y(P), y(T), y(B)}) \Big|_{y(P)}, \\ C_B &= -\frac{2D_L}{x(P)} y(P) \mathcal{L}'_3(y|_{y(P), y(T), y(B)}) \Big|_{y(P)}, \\ C_U &= -\frac{2D_L}{x(P)} z(P) \mathcal{L}'_2(z|_{z(P), z(U), z(D)}) \Big|_{z(P)}, \\ C_D &= -\frac{2D_L}{x(P)} z(P) \mathcal{L}'_3(z|_{z(P), z(U), z(D)}) \Big|_{z(P)}, \\ C_P &= -\frac{2D_L}{x(P)} \left(1 + y(P) \mathcal{L}'_1(y|_{y(P), y(T), y(B)}) \Big|_{y(P)} \right. \\ &\quad \left. + z(P) \mathcal{L}'_1(z|_{z(P), z(U), z(D)}) \Big|_{z(P)} \right). \end{aligned}$$

Once $V(R^*)$ is expressed in terms of the potentials of P and its neighbors, Laplace's equation can be enforced at boundary node P as normal 7-stencil equation. Since no additional neighbors is involved in the calculation of node P 's potential, there will be no changes in bandwidth and number of non-zero entries in the matrix.

For other faces of the numerical region, the same approach is used. Similar results can be obtained for the faces in the y and z directions.

3.3.2 Absorbing Boundary Condition (ABC2)

Using the chain rule, we can rewrite Eqn (3.7) as

$$\begin{aligned} & \xi_x \frac{\partial V}{\partial x} + \xi_y \frac{\partial V}{\partial y} + \xi_z \frac{\partial V}{\partial z} + \xi_{xx} \frac{\partial^2 V}{\partial x^2} + \xi_{yy} \frac{\partial^2 V}{\partial y^2} + \xi_{zz} \frac{\partial^2 V}{\partial z^2} + \xi_{xy} \frac{\partial^2 V}{\partial x \partial y} \\ & + \xi_{xz} \frac{\partial^2 V}{\partial x \partial z} + \xi_{yz} \frac{\partial^2 V}{\partial y \partial z} + \xi_0 V = 0, \end{aligned} \quad (3.12)$$

where

$$\begin{aligned} \xi_x &= \frac{-4x(x^2 + y^2)}{x^2 + y^2 + z^2}, \\ \xi_y &= \frac{-4y(x^2 + y^2)}{x^2 + y^2 + z^2}, \\ \xi_z &= \frac{-4z(x^2 + y^2)}{x^2 + y^2 + z^2}, \\ \xi_{xx} &= \frac{z^2 x^2}{x^2 + y^2 + z^2} + y^2, \\ \xi_{yy} &= \frac{z^2 y^2}{x^2 + y^2 + z^2} + x^2, \\ \xi_{zz} &= \frac{(x^2 + y^2)^2}{x^2 + y^2 + z^2}, \\ \xi_{xy} &= \frac{-2xy(x^2 + y^2)}{x^2 + y^2 + z^2}, \\ \xi_{xz} &= \frac{-2xz(x^2 + y^2)}{x^2 + y^2 + z^2}, \\ \xi_{yz} &= \frac{-2yz(x^2 + y^2)}{x^2 + y^2 + z^2}, \\ \xi_0 &= \frac{-2(x^2 + y^2)}{x^2 + y^2 + z^2}. \end{aligned}$$

Substituting the V_x from Eqn (3.12) into Eqn (3.9) gives

$$\begin{aligned}
V(R^*) = V(L) - \frac{2D_L}{\xi_x} & \left[\xi_y \frac{\partial V(P)}{\partial y} + \xi_z \frac{\partial V(P)}{\partial z} + \xi_{xx} \frac{\partial^2 V(P)}{\partial x^2} \right. \\
& + \xi_{yy} \frac{\partial^2 V(P)}{\partial y^2} + \xi_{zz} \frac{\partial^2 V(P)}{\partial z^2} + \xi_{xy} \frac{\partial^2 V(P)}{\partial x \partial y} \\
& \left. + \xi_{xz} \frac{\partial^2 V(P)}{\partial x \partial z} + \xi_{yz} \frac{\partial^2 V(P)}{\partial y \partial z} + \xi_0 V(P) \right]. \tag{3.13}
\end{aligned}$$

The approximations for $\partial V(P)/\partial y$ and $\partial V(P)/\partial z$ are the same as stated before. The second order derivative $\partial^2 V(P)/\partial y^2$ can be written in terms of a second derivative of the Lagrange interpolating coefficient function \mathcal{L} ,

$$\frac{\partial^2 V(P)}{\partial y^2} = \mathcal{L}''(y|_{y(P), y(T), y(B)}) \Big|_{y(P)} \cdot [V(P), V(T), V(B)]^T.$$

The derivative $\partial^2 V(P)/\partial z^2$ has a similar expression. However since $V(R^*)$ is unknown, $\partial^2 V(P)/\partial x^2$ cannot be approximated using the central difference. Instead, we employ a second-order accurate one-sided difference approximation,

$$V_{xx}(P) = \mathcal{L}''(x|_{x(P), x(L), x(LL)}) \Big|_{x(P)} \cdot [V(P), V(L), V(LL)]^T.$$

By applying the same concept for $V_{xy}(P)$, we obtain

$$V_{xy}(P) = \mathcal{L}'(x|_{x(P), x(L), x(LL)}) \Big|_{x(P)} \cdot \left[\frac{\partial V(P)}{\partial y}, \frac{\partial V(L)}{\partial y}, \frac{\partial V(LL)}{\partial y} \right]^T,$$

where

$$\frac{\partial V(L)}{\partial y} = \mathcal{L}'(y|_{y(L), y(TL), y(BL)}) \Big|_{y(L)} \cdot [V(L), V(TL), V(BL)]^T,$$

and

$$\begin{aligned} \frac{\partial V(LL)}{\partial y} &= \mathcal{L}'(y|_{y(LL), y(TLL), y(BLL)}) \Big|_{y(LL)} \\ &\quad \cdot [V(LL), V(TLL), V(BLL)]^T. \end{aligned}$$

Equations for $\partial^2 V(P)/\partial x \partial z$ and $\partial^2 V(P)/\partial y \partial z$ are similar. If we rewrite the coefficients of the related first order and second order derivative terms in the vector forms

$$\begin{aligned} [A_1, A_2, A_3] &= \mathcal{L}'(y|_{y(P), y(T), y(B)}), \\ [B_1, B_2, B_3] &= \mathcal{L}'(y|_{y(L), y(TL), y(BL)}), \\ [C_1, C_2, C_3] &= \mathcal{L}'(y|_{y(LL), y(TLL), y(BLL)}), \\ [D_1, D_2, D_3] &= \mathcal{L}'(z|_{z(P), z(U), z(D)}), \\ [E_1, E_2, E_3] &= \mathcal{L}'(z|_{z(L), z(UL), z(DL)}), \\ [F_1, F_2, F_3] &= \mathcal{L}'(z|_{z(LL), z(ULL), z(DLL)}), \\ [G_1, G_2, G_3] &= \mathcal{L}'(z|_{z(T), z(UT), z(DT)}), \\ [H_1, H_2, H_3] &= \mathcal{L}'(z|_{z(B), z(UB), z(DB)}), \\ [I_1, I_2, I_3] &= \mathcal{L}'(x|_{x(P), x(L), x(LL)}), \\ [J_1, J_2, J_3] &= \mathcal{L}''(y|_{y(P), y(T), y(B)}), \\ [K_1, K_2, K_3] &= \mathcal{L}''(z|_{z(P), z(U), z(D)}), \\ [L_1, L_2, L_3] &= \mathcal{L}''(x|_{x(P), x(L), x(LL)}), \end{aligned}$$

the three mixed derivative terms can be written in the compact forms

$$V_{xy}(P) = \begin{bmatrix} I_1 & I_2 & I_3 \end{bmatrix} \begin{bmatrix} A_1V(P) + A_2V(T) + A_3V(B) \\ B_1V(L) + B_2V(TL) + B_3V(BL) \\ C_1V(LL) + C_2V(TLL) + C_3V(BLL) \end{bmatrix},$$

$$V_{xz}(P) = \begin{bmatrix} I_1 & I_2 & I_3 \end{bmatrix} \begin{bmatrix} D_1V(P) + D_2V(U) + D_3V(D) \\ E_1V(L) + E_2V(UL) + E_3V(DL) \\ F_1V(LL) + F_2V(ULL) + F_3V(DLL) \end{bmatrix},$$

$$V_{yz}(P) = \begin{bmatrix} A_1 & A_2 & A_3 \end{bmatrix} \begin{bmatrix} D_1V(P) + D_2V(U) + D_3V(D) \\ G_1V(T) + G_2V(UT) + G_3V(DT) \\ H_1V(B) + H_2V(UB) + H_3V(DB) \end{bmatrix}.$$

With the approximation of all derivatives, we can rewrite Eqn (3.13) as

$$\begin{aligned} V(R^*) = & C_PV(P) + C_TV(T) + C_BV(B) + C_UV(U) \\ & + C_DV(D) + C_LV(L) + C_{TL}V(TL) + C_{BL}V(BL) \\ & + C_{UL}V(UL) + C_{DL}V(DL) + C_{LL}V(LL) \\ & + C_{TLL}V(TLL) + C_{BLL}V(BLL) + C_{ULL}V(ULL) \\ & + C_{DLL}V(DLL) + C_{UT}V(UT) + C_{DT}V(DT) \\ & + C_{UB}V(UB) + C_{DB}V(DB). \end{aligned}$$

where

$$\begin{aligned}
C_P &= \xi'_y A_1 + \xi'_z D_1 + \xi'_{yy} J_1 + \xi'_{zz} K_1 + \xi'_{xx} L_1 + \xi'_{xy} I_1 \cdot A_1 + \xi'_{xz} I_1 \cdot D_1 + \xi'_{yz} A_1 \cdot D_1 + \xi'_0, \\
C_T &= \xi'_y A_2 + \xi'_{yy} J_2 + \xi'_{xy} I_1 \cdot A_2 + \xi'_{yz} A_2 \cdot G_1, \\
C_B &= \xi'_y A_3 + \xi'_{yy} J_3 + \xi'_{xy} I_1 \cdot A_3 + \xi'_{yz} A_3 \cdot H_1, \\
C_U &= \xi'_z D_2 + \xi'_{zz} K_2 + \xi'_{xz} I_1 \cdot D_2 + \xi'_{yz} A_1 \cdot D_2, \\
C_D &= \xi'_z D_3 + \xi'_{zz} K_3 + \xi'_{xz} I_1 \cdot D_3 + \xi'_{yz} A_1 \cdot D_3, \\
C_L &= 1 + \xi'_{xx} L_2 + \xi'_{xy} I_2 \cdot B_1 + \xi'_{xz} I_2 \cdot E_1, \\
C_{TL} &= \xi'_{xy} I_2 \cdot B_2, \\
C_{BL} &= \xi'_{xy} I_2 \cdot B_3, \\
C_{UL} &= \xi'_{xz} I_2 \cdot E_2, \\
C_{DL} &= \xi'_{xz} I_2 \cdot E_3, \\
C_{LL} &= \xi'_{xx} L_3 + \xi'_{xy} I_3 \cdot C_1 + \xi'_{xz} I_3 \cdot F_1, \\
C_{TLL} &= \xi'_{xy} I_3 \cdot C_2, \\
C_{BLL} &= \xi'_{xy} I_3 \cdot C_3, \\
C_{ULL} &= \xi'_{xz} I_3 \cdot F_2, \\
C_{DLL} &= \xi'_{xz} I_3 \cdot F_3, \\
C_{UT} &= \xi'_{yz} A_2 \cdot G_2, \\
C_{DT} &= \xi'_{yz} A_2 \cdot G_3, \\
C_{UB} &= \xi'_{yz} A_3 \cdot H_2, \\
C_{DB} &= \xi'_{yz} A_3 \cdot H_3,
\end{aligned}$$

and the normalized coefficient

$$\begin{aligned}
\xi'_y &= -2D_L \frac{\xi_y}{\xi_x} \\
\xi'_z &= -2D_L \frac{\xi_y}{\xi_x} \\
\xi'_{xx} &= -2D_L \frac{\xi_{xx}}{\xi_x} \\
\xi'_{yy} &= -2D_L \frac{\xi_{yy}}{\xi_x} \\
\xi'_{zz} &= -2D_L \frac{\xi_{zz}}{\xi_x} \\
\xi'_{xy} &= -2D_L \frac{\xi_{xy}}{\xi_x} \\
\xi'_{xz} &= -2D_L \frac{\xi_{xz}}{\xi_x} \\
\xi'_{yz} &= -2D_L \frac{\xi_{yz}}{\xi_x} \\
\xi'_0 &= -2D_L \frac{\xi_0}{\xi_x}
\end{aligned}$$

Once $V(R^*)$ is determined, Laplace's equation can be applied to the boundary node P . Since more neighbor nodes are needed for approximation of $V(P)$, both the bandwidth and non-zero entries will increase.

3.3.3 Implement ABC1 with Artificial Layer (AL)

To make ABC easy to implement, we develop a method to convert ABC1 into an artificial dielectric layer. It can be applied to any existing field solver without implementation modification by adding a layer of dielectric at the outer boundary. The artificial dielectric is non-homogeneous, and has dependency on the multi-layer dielectric constants in the numerical region.

In [20], an energy equivalent dielectric layer of open boundary was derived with the assumption that the dielectric is homogeneous. Since their boundary condition

has lower accuracy than ABC1, and their approach is different, we independently derive our more accurate artificial layer as follows.

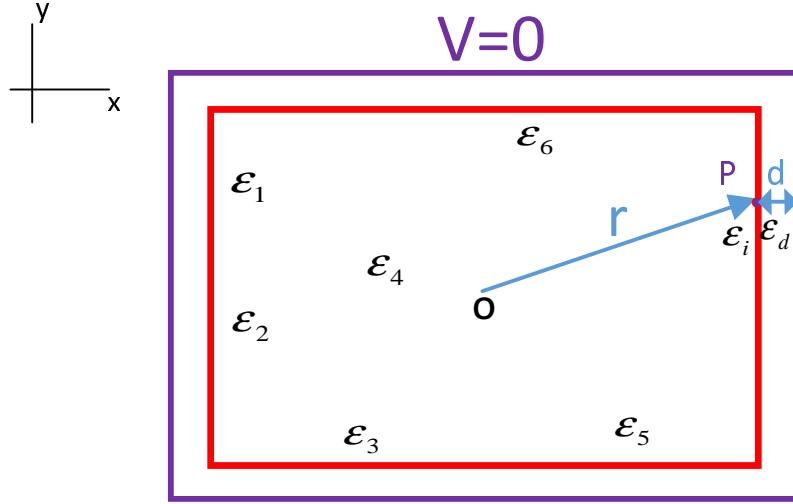


Figure 3.3: Top view of absorbing boundary condition with an artificial layer of thickness d . The boundary of the numerical region is marked with red color while the exterior Dirichlet boundary is marked with purple color.

Fig. 3.3 is the general setting of the problem, where the radial distance of node P on the boundary is r and the dielectric constant at node P is ϵ_i . We add an artificial dielectric layer of thickness d and permittivity ϵ_d outside of the numerical region. We also add a Dirichlet boundary $V = 0$ on the exterior of this artificial layer. Our goal is to enforce ABC1, which requires the following condition be met at P ,

$$\frac{\partial V}{\partial r} = -\frac{V}{r}. \quad (3.14)$$

Assume without loss of generality P is on the $+X$ face boundary with coordinate

x , then the electric field at node P inside of the numerical region is

$$\left. \frac{\partial V}{\partial n} \right|_{P_-} = \left. \frac{\partial V}{\partial r} \frac{\partial r}{\partial x} \right|_{P_-} = -\frac{V_P}{r} \frac{x}{r}. \quad (3.15)$$

On the other hand, due to the Dirichlet boundary and thickness d , the normal electric field at node P outside of the numerical region but inside of the artificial layer is

$$\left. \frac{\partial V}{\partial n} \right|_{P_+} = \frac{0 - V_P}{d}. \quad (3.16)$$

Finally, we define the relative permittivity of the artificial layer around P as

$$\epsilon_d = \epsilon_i \frac{d \cdot x}{r^2},$$

due to the continuity of normal electric flux density on P ,

$$\epsilon_i \left. \frac{\partial V}{\partial n} \right|_{P_-} = \epsilon_d \left. \frac{\partial V}{\partial n} \right|_{P_+}. \quad (3.17)$$

Thus ABC1 Eqn (3.14) is automatically satisfied.

We also call artificial layer satisfying ABC1 as AL for simplicity in the later sections. It is worth noting that the artificial layer is distinct from the static perfectly matched layer approaches [28][29] where the artificial layer is anisotropic.

3.4 Application to other Methods

The artificial layer approach can be applied to any field solver without modification to the field solver. In this section, we describe how ABC1 and ABC2 can be directly applied to other methods.

3.4.1 Finite Element Method

Multiplying the Laplace's equation by a testing function ψ , and integrating over the region Ω , we get

$$\int_{\Omega} \nabla \psi \cdot \nabla V dV = \int_{\Gamma} \psi \nabla V dS, \quad (3.18)$$

by applying the Green's first identity [30]. The right hand side of Eqn (3.18) represents the flux contribution from the outer boundary which is zero when Neumann boundary condition is applied.

Suppose the whole numerical region Ω will be discretized into tetrahedral or hexahedron element. The weak formulation for the inner elements is the same as Eqn (3.18) except the right hand side is zero. The absorbing boundary condition needs to impose to the outermost boundary elements. For the elements touching the $+X$ face of the region, ABC1 can be incorporated into Eqn (3.18) to yield

$$\int_{\Omega} \nabla \psi \cdot \nabla V dV = \int_{\Gamma} \psi \frac{\partial V}{\partial x} dydz,$$

where

$$\begin{aligned} \frac{\partial V}{\partial x} = \frac{1}{2D_L} & \left((C_L - 1)V(L) + C_TV(T) + C_BV(B) + C_UV(U) \right. \\ & \left. + C_DV(D) + C_PV(P) \right). \end{aligned}$$

Similar conclusions can be made for outermost elements with other outward normal directions. When ABC2 is applied, $\partial V/\partial x$ will be replaced with high order terms derived from Eqn (3.12).

3.4.2 Floating Random Walk

Floating Random Walk (FRW) method [31] is a Monte Carlo method for integration and capacitance calculation through random walks in dielectric space. The random walk path starts at any point on the Gaussian surface of conductor at 1 volt, and continues to the next point with the probability defined by the Green's function within the transition domain bounded by the nearest conductor. If the path arrives any conductor, it will stop.

Since the walk domain is bounded, when it arrives at the boundary, its walking path is restricted to remain inside, which means the outward walking probability is 0. Thus, a truncation error is introduced.

ABC1 can be used in FRW to reduce such truncation error. For instance, when a random walk path touches the boundary node P as shown in Fig. 3.2, it can continue its path through either node U , D , B , T , L or R^* with probability p_U , p_D , p_B , p_T , p_L and p_{R^*} , assuming it is a fixed random walk for node P . Combining Eqn (3.11) with conservation law, it is easy to find

$$V(P) = \frac{p_L + p_{R^*}C_L}{1 - p_{R^*}C_P}V(L) + \frac{p_T + p_{R^*}C_T}{1 - p_{R^*}C_P}V(T) + \frac{p_B + p_{R^*}C_B}{1 - p_{R^*}C_P}V(B) \\ + \frac{p_U + p_{R^*}C_U}{1 - p_{R^*}C_P}V(U) + \frac{p_D + p_{R^*}C_D}{1 - p_{R^*}C_P}V(D).$$

The coefficients for $V(L)$, $V(T)$, \dots , $V(D)$ are the probabilities that the random walk will move to L , T , \dots , D , respectively. This will guarantee that the random walk in bounded domain achieving the same accuracy as walking in infinite domain.

3.5 Experiments

We performed three experiments to verify the accuracy and efficiency of the absorbing boundary conditions. We used direct solver CHOLMOD [32] to solve the

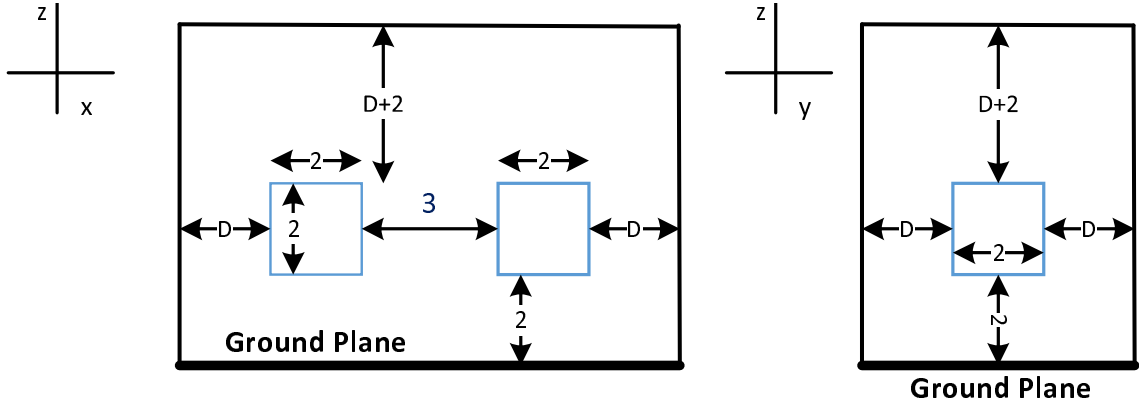


Figure 3.4: Experiment I: Two conductors in free space.

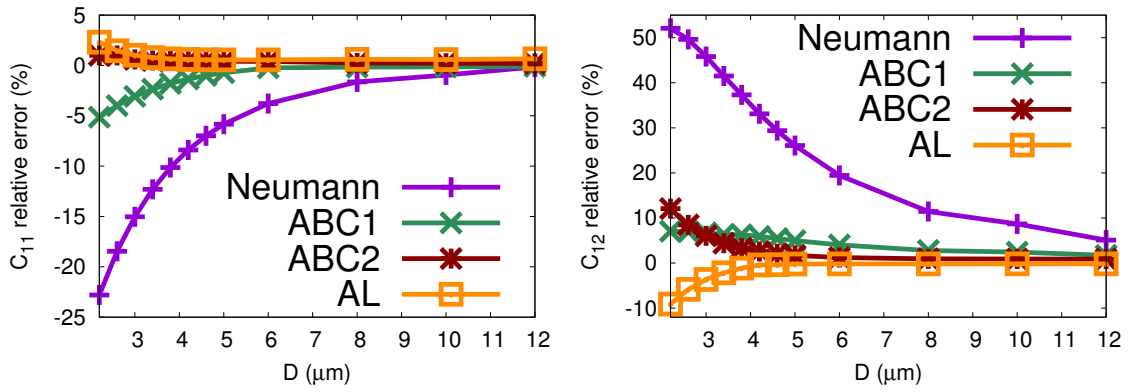


Figure 3.5: Accuracy of total capacitance and coupling capacitance as a function of D for Experiment I shown in Fig. 3.4.

sparse linear system to avoid potential error caused by the iterative solver. All golden reference results are generated from commercial field solver.

3.5.1 Conductors in Free Space

The setting of the experiment I is shown in Fig. 3.4, and the units are micro metres.

The external boundary is at a distance D to the conductors in $-x$, $+x$, $-y$, $+y$

direction, and $D+2$ in $+z$ directions. In this experiment, uniform mesh is utilized. The reference result is total capacitance (C_{11}) 0.201 fF and coupling capacitance (C_{12}) -0.0382 fF when $D = 20 \mu\text{m}$.

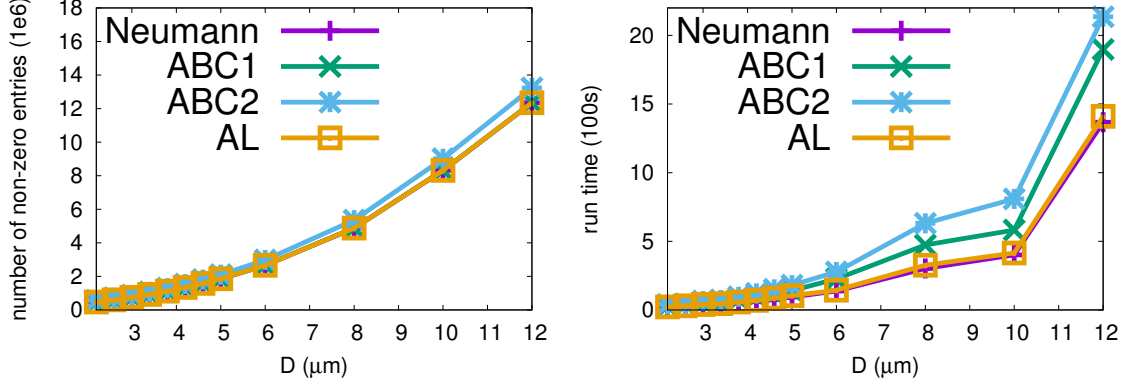


Figure 3.6: Number of non-zero entries and run time as a function of distance D for Experiment I shown in Fig. 3.4.

The relative error with respect to distance is demonstrated in Fig. 3.5. When ABC1 is applied and D is $8 \mu\text{m}$, the result for C_{11} is 0.201 fF, and for C_{12} is -0.0392 fF. They differ from the reference result by only -0.16% and 2.78% respectively. When ABC2 is applied, $D = 4.2 \mu\text{m}$ gives comparable accuracy. While with AL, even $D = 3.8 \mu\text{m}$ provides similar accuracy. However, when Neumann boundary condition is applied, to achieve similar accuracy, D is at least $12 \mu\text{m}$. Thus, ABC1, ABC2 and AL have higher accuracy than Neumann boundary condition for the same D .

The comparisons involve not only the accuracy, but also the number of non-zero entries and run time, which are shown in Fig. 3.6. Obviously the number of entries for both Neumann boundary condition, ABC1 and AL are almost the same. While ABC2 has about 15% more non-zero entries in the matrix.

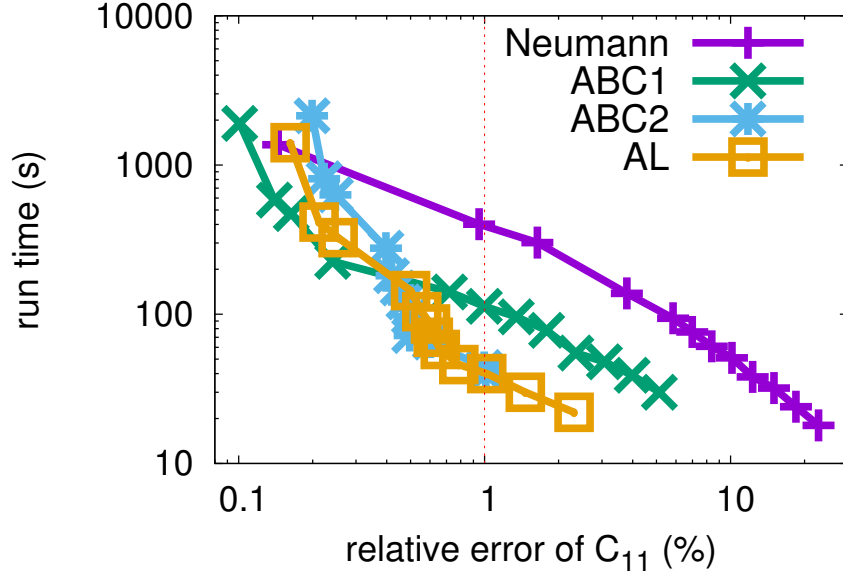


Figure 3.7: Run time as a function of relative error for total capacitance C_{11} in Experiment I.

ABC1 is 50% slower than Neumann boundary condition for the same distance D due to the non-symmetry of the matrix, while ABC2 is even worse. However, for the same accuracy, ABC1 and ABC2 use far less time. Since the matrix for AL is still sparse and symmetric, the time increase is less than 10% compared with Neumann boundary condition for the same D . As shown in Fig. 3.7, when the relative error of C_{11} is around 1%, the run time for ABC1, ABC2 and AL is 113s, 44s and 40s respectively, while 401s for Neumann boundary condition. Therefore, ABCs have four to ten times speed up to achieve the same accuracy. Since AL requires fewer computational cost to achieve the similar accuracy as ABC1 and ABC2, it is a preferable choice.

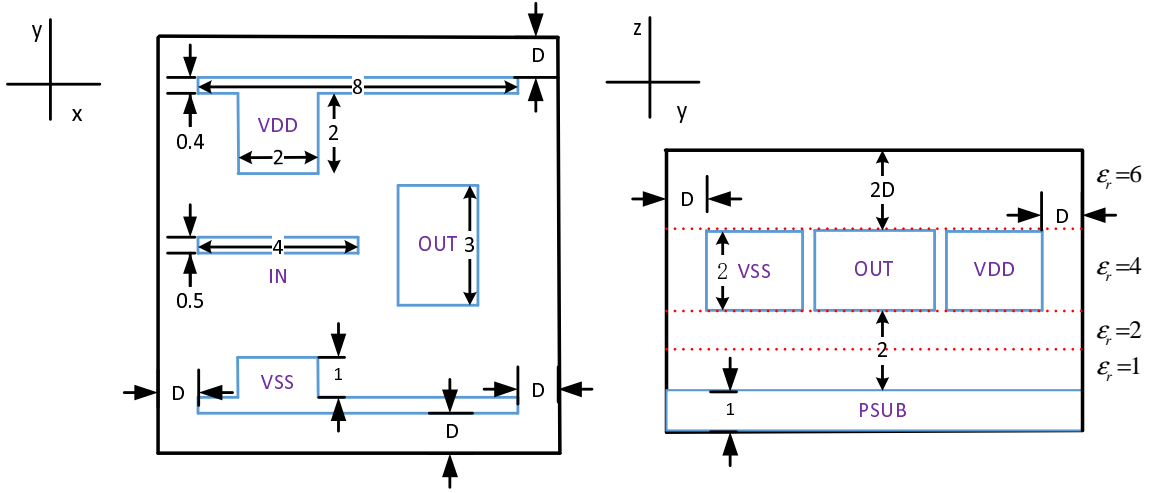


Figure 3.8: Experiment II: Part of an inverter in multi-dielectric layers.

3.5.2 Conductors in Multi-layer Dielectric

The second experiment is depicted in Fig. 3.8 with all dimensional details. There are four nets IN, OUT, VDD, VSS as well as PSUB (P-substrate) as the ground. It is worth noting that the numerical region contains four layers of dielectric material. In this test, we measure the total capacitance of VDD and the coupling capacitance between VDD and other nets. We call the total capacitance of VDD as C_{11} , while the coupling capacitance with IN, OUT, VSS as C_{12} , C_{13} and C_{14} , respectively. We obtain the reference result from the domain of $200\text{ }\mu\text{m} \times 200\text{ }\mu\text{m} \times 200\text{ }\mu\text{m}$ with about 10 million grid points, where $C_{11} = 1.36\text{fF}$, $C_{12} = -0.253\text{fF}$, $C_{13} = -0.229\text{fF}$, $C_{14} = -0.0538\text{fF}$. Since the uniform mesh will yield an incredibly large matrix for this complicated case, which makes it intractable, we employ the adjusted mesh generated by the commercial package to reduce the degree of freedom while maintaining the accuracy.

All results are in Fig. 3.9, Fig. 3.10 and Fig. 3.11. As shown, the performance

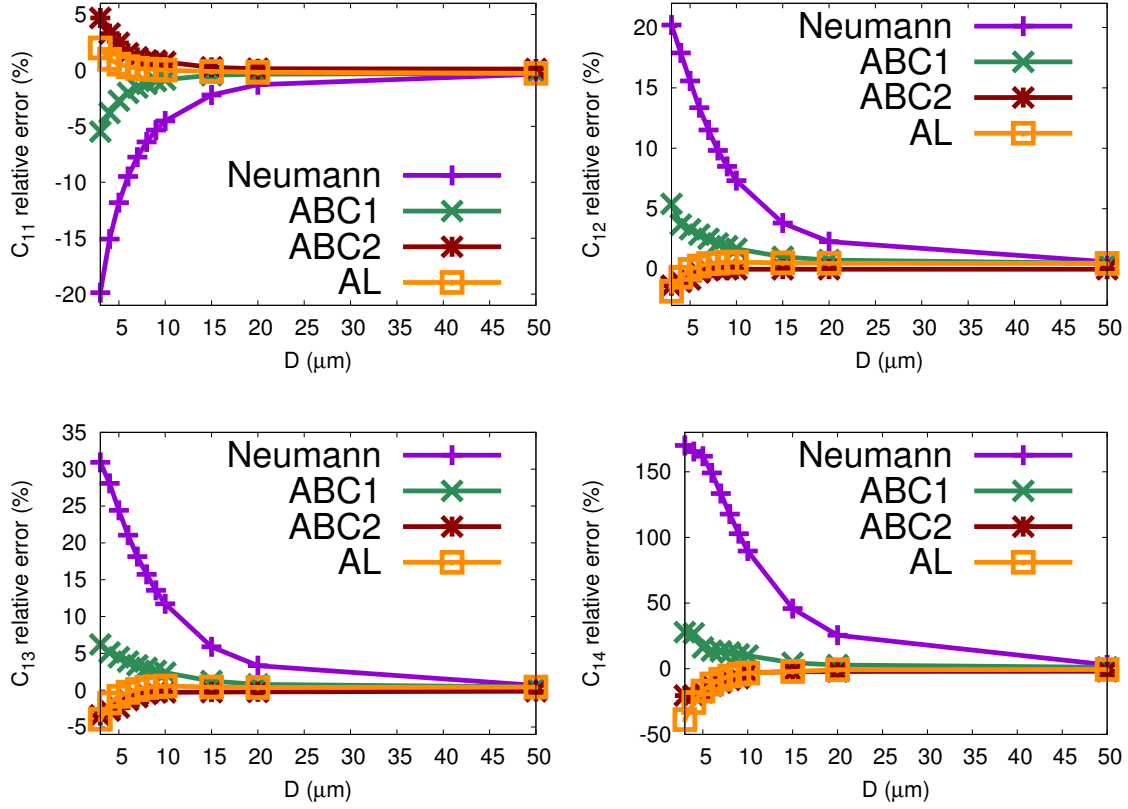


Figure 3.9: Accuracy of total capacitance and coupling capacitance as a function of D for Experiment II shown in Fig. 3.8.

comparisons in terms of accuracy, non-zero entries as well as run time, have patterns similar to experiment I. For example, when $D = 10 \mu\text{m}$ and ABC2 is utilized, the capacitance results have relative errors of 0.75%, -0.01%, -0.29%, -3.00% respectively, and it requires 356s to simulate. When AL is applied, the comparable accuracy can be obtained with only 253s. The situation with Neumann boundary condition is worse, the error is as high as 89.59%. To achieve the same accuracy, the numerical region of Neumann boundary condition has to be extended to $D = 50 \mu\text{m}$ at least, which requires 1056s to simulate, almost four times run time as AL.

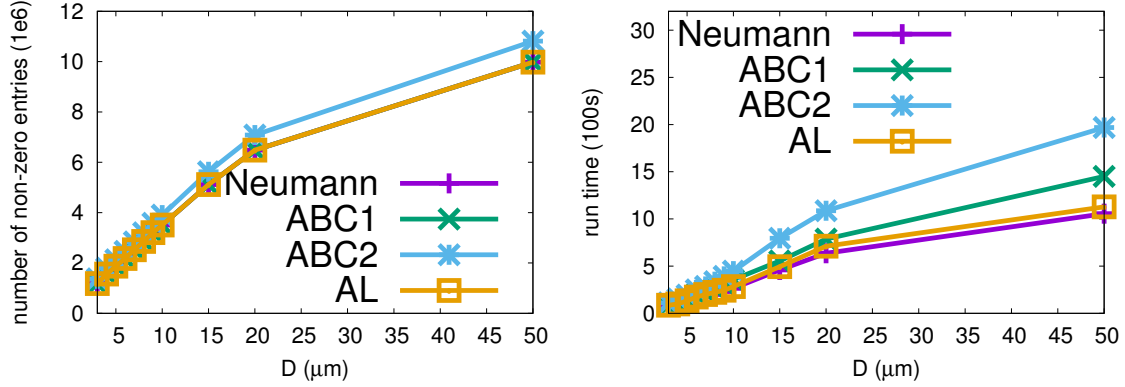


Figure 3.10: Number of non-zero entries and run time as a function of D for Experiment II shown in Fig. 3.8.

3.5.3 Conductors in SOI

The third experiment is an SOI case. SOI uses a layered silicon insulator-silicon-substrate in place of conventional silicon substrates, the ground is far away. When ground is far away, traditional boundary will require much greater region. In the experiment shown in Fig. 3.12, the simulation result with boundary element method (BEM) is taken as the reference, where $C_{AA} = 0.179\text{fF}$, $C_{AB} = -0.0457\text{fF}$, $C_{AC} = -0.0571\text{fF}$. To achieve the same accuracy, the numerical region for FDM with Neumann boundary condition should be at least $1000\mu\text{m} \times 1000\mu\text{m} \times 1000\mu\text{m}$ with 10 million grid points. We employ AL to take the far away ground into consideration. Fig. 3.13 show better accuracy of AL compared with Neumann boundary condition. When the distance between the conductors and the outer boundary $D = 16\mu\text{m}$, the results of applying AL have errors of 1.43%, -1.44%, -2.64% respectively. To obtain similar accuracy, traditional approach will require a numerical region of at least $80\mu\text{m} \times 80\mu\text{m} \times 80\mu\text{m}$ with the same mesh density. From Fig. 3.14, we can find that to achieve 3.6% error for C_{AB} for instance, Neumann boundary condition requires

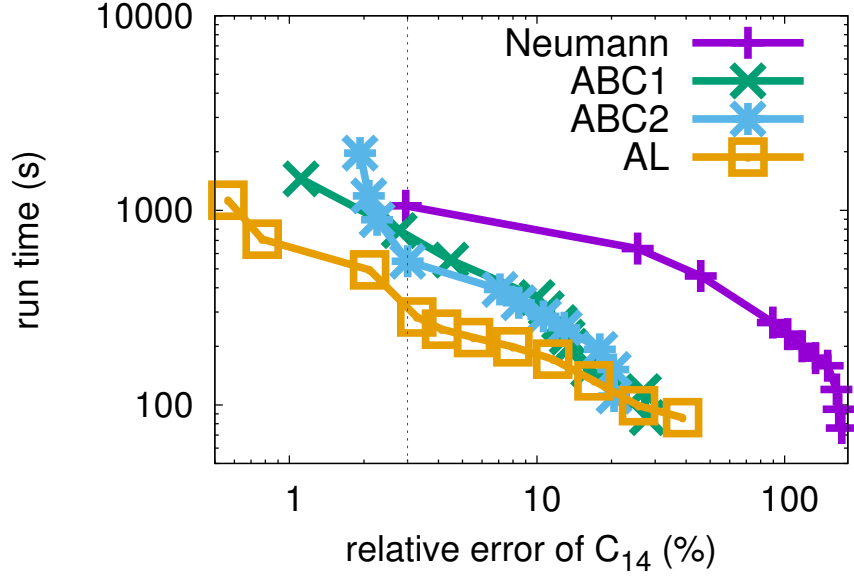


Figure 3.11: Run time as a function of relative error for coupling capacitance C_{14} in Experiment II.

run time of 2388s while AL only requires 158s, which indicates more than ten times speed up when AL is applied.

3.6 Conclusion

Motivated by the goal of reducing capacitance extraction error due to the inaccurate boundary conditions on the periphery of the numerical region modeling circuits and interconnects, we have developed a class of absorbing boundary conditions. Conditions ABC1 and ABC2 are approximated in Cartesian coordinates with first and second order numerical derivatives as well as mixed partial derivatives. They are utilized to approximate the inaccessible neighbors of the boundary nodes. The conservation law was then rewritten with all neighbors taken into consideration at boundary nodes. In addition, a simple artificial layer approach was introduced

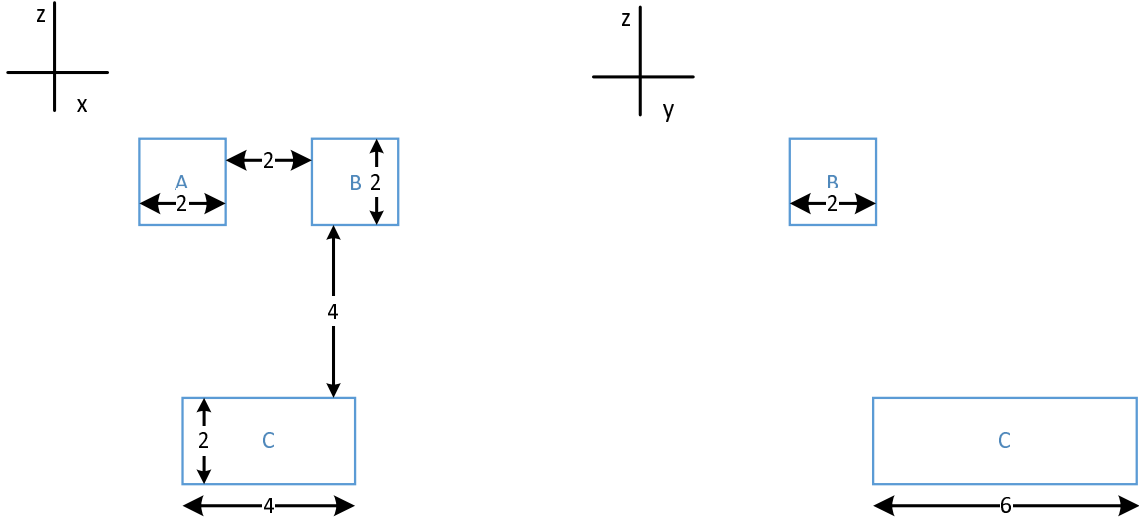


Figure 3.12: Experiment III: Three conductors in SOI.

based on the continuity of normal electric flux density. Absorbing boundary conditions for parasitic capacitance extraction, can be implemented either by rewriting the linear equation for boundary nodes with explicit expression of the neighboring node outside of the simulation region or by artificially adding a thin lossy layer outside of the region. The absorbing layer is terminated with the $V=0$ Dirichlet boundary condition.

Simulations have demonstrated the superior performance of the proposed absorbing conditions over the Neumann boundary condition in accuracy and efficiency. Since the Neumann boundary condition is widely used in 3-D volume based solvers, the absorbing boundary conditions may improve the accuracy and efficiency of a large class of field solvers.

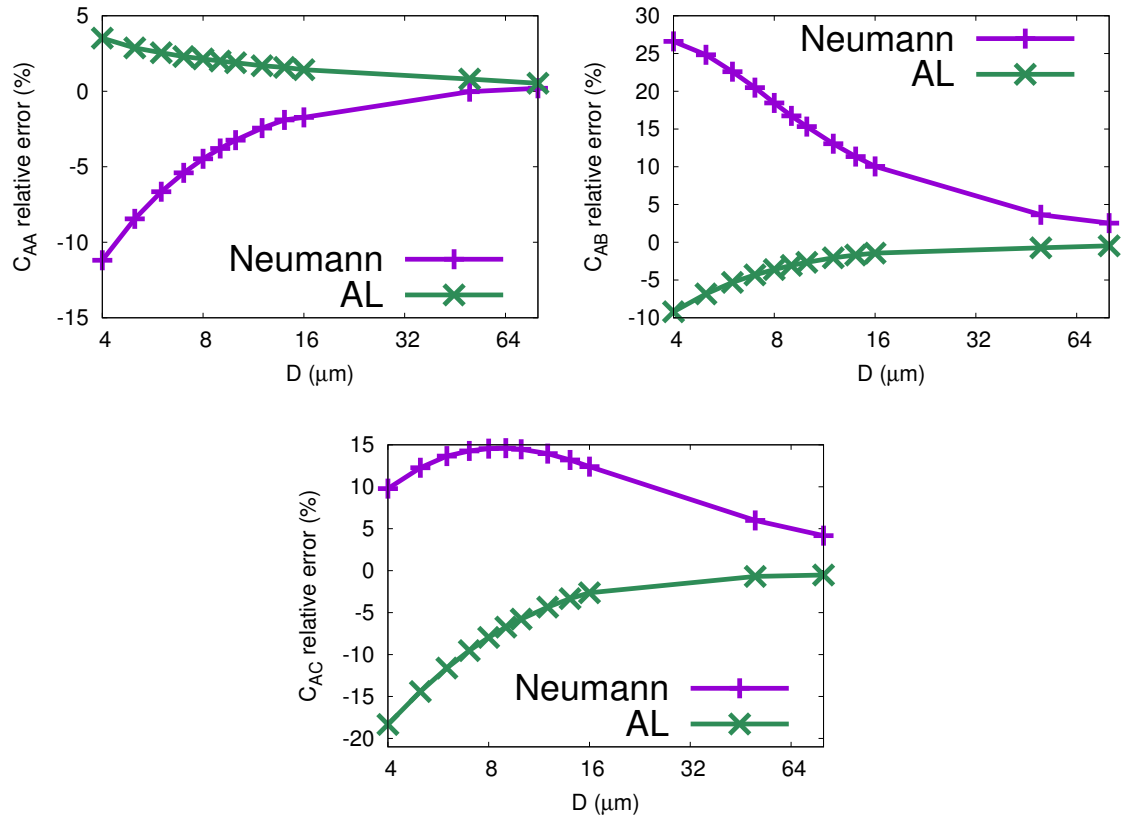


Figure 3.13: Accuracy of total capacitance and coupling capacitance as a function of D for Experiment III shown in Fig. 3.12.

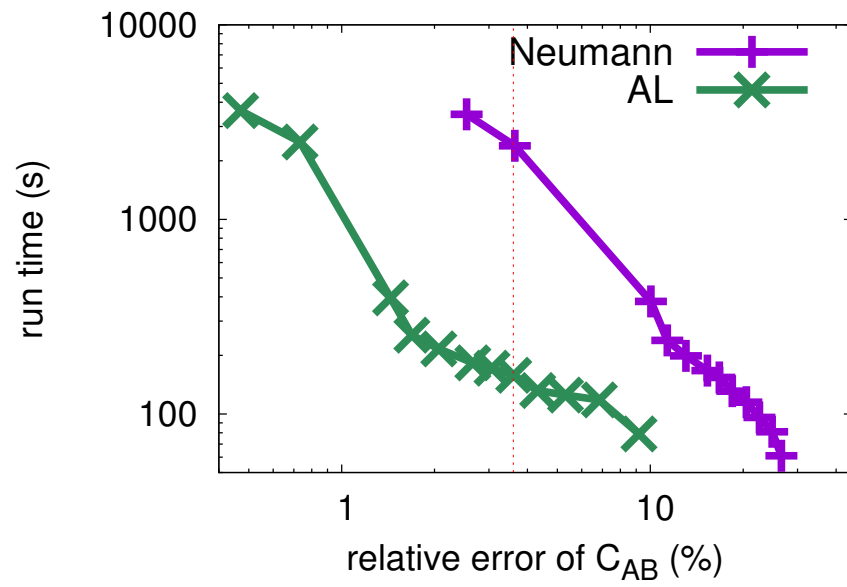


Figure 3.14: Run time as a function of relative error for coupling capacitance C_{AB} in Experiment III.

4. RESISTANCE EXTRACTION FOR ANISOTROPIC LAYER

4.1 Background

Finite difference method (FDM), Finite element method (FEM) and Finite volume method (FVM) belong to the class of volume methods, and the discretization in the simulation domain is prerequisite. The classic FDM [33] is a special case of FVM, where the volume for any node is a rectangle defined by connecting perpendicular bisectors (PEBI) of the edges impinging on this node for 2D case. However, FDM can only provide poor accuracy to the case with non-orthogonal boundary since grids are confined to coordinate lines only, which makes it hard to converge and can only use high degree of freedom (DOF) to compensate. Both FEM and FVM are easy to apply to the non-orthogonal grids. The traditional FEM [34] assumes continuous or weakly continuous approximation spaces and ask for element integrals of the weak form to be satisfied. This method is not conservative, thus often struggle with stability for discontinuous processes, which results in low accuracy for coarse mesh. Besides, FEM is sensitive to the distorted mesh and produces poor result for low quality mesh. FVM uses piecewise constant approximation spaces and requires integrals against piecewise constant test functions to be satisfied. Thus exact conservation statements is guaranteed which makes FVM more robust. This is very important for the differential equations with hyperbolic feature or fluid with phase transition. The feature comparison of these three numerical methods is summarized in Table. 4.1.

Both FDM and FEM are widely used for the resistance extraction problem. FVM has already been widely used in the computational fluid dynamics [35], and petroleum reservoir simulation [36]. However, FVM has never been introduced to this field. In

addition, when FEM and FVM are used, cartesian grid can not be maintained. Instead, triangular or quadrilateral mesh is prerequisite, and re-meshing is needed whenever there are small changes in any specific domain. Such expensive costs make both FEM and FVM hard to apply to practical problems in industry. The Immersed Finite Element Method (IFEM) is a modified FEM working on cartesian grid with non-orthogonal interface problem. We will introduce both FVM and IFEM in this chapter, and evaluate their performance for resistance extraction with anisotropic layer.

Table 4.1: Feature comparison of FDM, FEM and FVM.

Feature	FDM	FEM	FVM
non-rectilinear geometry	poor	good	good
easy to generate mesh	easy	hard	medium
conservation law	yes	no	yes
robustness	good	medium	good

In the classical resistance extraction problem, the potential follows the diffusion equation

$$\nabla \cdot \sigma \nabla V = 0, \quad (4.1)$$

where σ is the conductivity of the conductor. The goal of resistance extraction is to obtain the resistance with total flux at the Dirichlet boundary Γ

$$I = \int_{\Gamma} \sigma \frac{\partial V}{\partial n} = \frac{1}{R}. \quad (4.2)$$

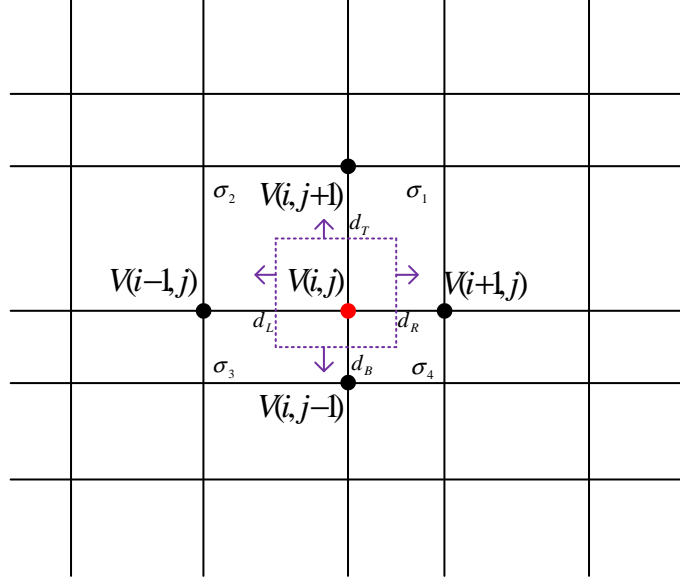


Figure 4.1: Five point stencil in 2D FDM.

4.2 Finite Difference Method

FDM is one of the simplest and of the oldest methods to solve differential equations in which finite differences are used to approximate the derivatives. To apply FDM, the domain is partitioned into small rectangular or cubic domain with cartesian grid, and the finite number of dependent variables are defined on each grid point. The partial derivatives at each grid point are approximated with first order differences. The first order difference or Euler expression is based on the Taylor's theorem. For instance, as shown in Fig. 4.1, grid node (i, j) has distance d_L , d_R , d_B and d_T with its neighboring nodes. The potential at the node $(i + 1, j)$ can be

written with taylor expansion

$$V(i+1, j) = V(i, j) + d_R V_x(i, j) + d_R^2 \frac{V_{xx}(i, j)}{2!} + \dots + d_R^{n-1} \frac{V_{n-1}(i, j)}{(n-1)!} + O(d_R^n),$$

assuming the distance d_R is small enough. We obtain

$$V(i+1, j) = V(i, j) + d_R V_x(i, j) + O(d_R^2).$$

Rearranging the equation and neglecting the $O(d_R^2)$ term will give the forward difference in x direction

$$V_x^+(i, j) = \frac{V(i+1, j) - V(i, j)}{d_R}.$$

Similarly, the backward difference in x direction is

$$V_x^-(i, j) = \frac{V(i, j) - V(i-1, j)}{d_L}.$$

The diffusion term at node (i, j) will be calculated at the closed integration surface which is formed by the PEBI of the edges between this node and its neighboring nodes. The integration surface is marked with purple color in Fig .4.1. With the assumption that each face on the integration surface is identified as \vec{A}_i , the original diffusion equation at node (i, j) can be written as

$$\sum_{i=1}^4 \sigma \nabla V \cdot \vec{A}_i = 0.$$

The flux between $(i - 1, j)$ and (i, j) through face \vec{A}_1 is

$$\sigma \nabla V \cdot \vec{A}_1 = \frac{V(i - 1, j) - V(i, j)}{d_L} \frac{\sigma_2 d_T + \sigma_3 d_B}{2}.$$

Similar results for flux at other faces. Therefore, the conservation law at node (i, j) can be written with five point stencil

$$\begin{aligned} & \frac{\sigma_2 d_T + \sigma_3 d_B}{2d_L} V(i - 1, j) + \frac{\sigma_1 d_T + \sigma_4 d_B}{2d_R} V(i + 1, j) \\ & + \frac{\sigma_3 d_L + \sigma_4 d_R}{2d_B} V(i, j - 1) + \frac{\sigma_2 d_L + \sigma_1 d_R}{2d_T} V(i, j + 1) \\ & - \left(\frac{\sigma_2 d_T + \sigma_3 d_B}{2d_L} + \frac{\sigma_1 d_T + \sigma_4 d_B}{2d_R} + \frac{\sigma_3 d_L + \sigma_4 d_R}{2} \frac{\sigma_2 d_L + \sigma_1 d_R}{2d_T} \right) V(i, j) = 0. \end{aligned}$$

4.3 Finite Volume Method

The term “finite volume” refers to the small volume surrounding each node point on a mesh. FVM is normally categorized as node centered (NC) and cell centered (CC) as shown in Fig. 4.2. In the NC-FVM, unknown quantities are stored in the mesh vertices, while solutions are defined at the center of the mesh cells with the primal cells serving as the control volume for the CC-FVM. Depending on whether the diffusion flux between two adjacent cell is two-cells stencil or not, FVM can be classified as two-point flux approximation (TPFA) and multi-point flux approximation (MPFA).

In [37], the standard finite volume algorithm utilizes the crude finite difference approximation to discretize the face fluxes, which makes it behave poorly in accuracy. For non-orthogonal mesh, the flux at the cell face is typically reconstructed from the gradient between neighboring cells and augmented with the face gradient [38][39]. There are two reconstruction scheme, either node-averaging face gradient [40] or

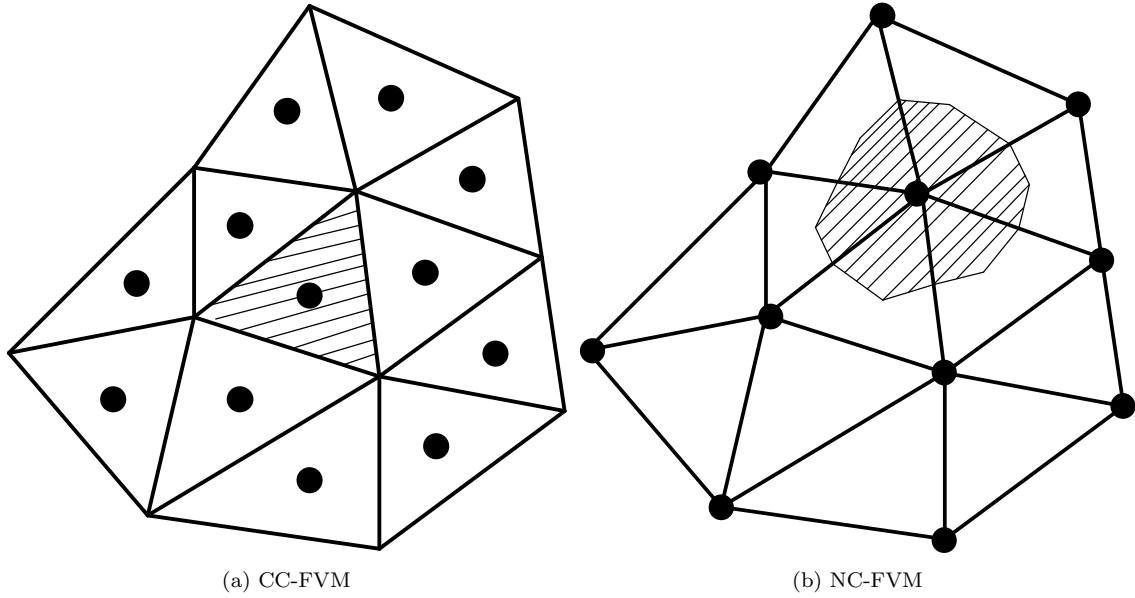


Figure 4.2: Mesh of CC-FVM and NC-FVM. The unknowns are associated to black nodes, with the control volume marked with shaded domain

least-squares scheme face gradient [41]. Both stencils are asymmetric, which result more computational cost and have difficulties to maintain accuracy on high-aspect-ratio curved grids [42].

Morel [43] developed a cell-centered diffusion scheme which treats rigorously material discontinuities and gives a second order accuracy. However, the drawbacks are there are cell-face unknowns in addition to the cell-centered unknowns and the global diffusion matrix is unsymmetric. Both the local flux mimetic finite difference methods [44] and the Multi-Point Flux Approximation (MPFA) [36][45][46] uses two degrees of freedom per edge to construct the local flux discretization, and has only cell-centered unknowns and a local stencil. The only disadvantage of MPFA is the diffusion matrix is asymmetric for the general quadrilaterals and triangles. In [47][48], another MPFA scheme was developed with symmetric sparse matrix. It has been

verified that it has second order accuracy for triangular grids, while almost second order accuracy for quadrilateral grids except for non smooth grids.

In this section, we employ TPFA and MPFA [47][48] to do resistance extraction. This is the first time to apply FVM to extraction problem. We provide detailed description of this method including the discretization of the local matrix and application to boundary conditions.

4.3.1 Two-Point Flux Approximation

4.3.1.1 Discretization Scheme

If we consider a general grid with two neighboring cells Ω^i and Ω^{i+1} separated by edge $[P_k, P_{k+1}]$ as depicted in Fig. 4.3, then the grid is orthogonal when there exists a point on the face, say $P_{k+\frac{1}{2}}$, such that $\overrightarrow{G_i P_{k+\frac{1}{2}}}$ and $\overrightarrow{G_{i+1} P_{k+\frac{1}{2}}}$ are orthogonal to the face where G_i and G_{i+1} are the cell centers. In such case, the flux between these two cells becomes one-dimensional problem.

With the assumption that the mean potential at the face is $\bar{V}_{k+\frac{1}{2}}$, we define the flux F_i at the interface due to the cell Ω_i with transmissibilities T_i satisfying

$$F_i = T_i(V_i - \bar{V}_{k+\frac{1}{2}}) = \frac{\sigma_i \left| \overrightarrow{p_k p_{k+1}} \right|}{\left| \overrightarrow{G_i P_{k+\frac{1}{2}}} \right|} (V_i - \bar{V}_{k+\frac{1}{2}}). \quad (4.3)$$

Similar definition for F_{i+1} with transmissibilities T_{i+1}

$$F_{i+1} = T_{i+1}(\bar{V}_{k+\frac{1}{2}} - V_{i+1}) = \frac{\sigma_{i+1} \left| \overrightarrow{p_k p_{k+1}} \right|}{\left| \overrightarrow{P_{k+\frac{1}{2}} G_{i+1}} \right|} (\bar{V}_{k+\frac{1}{2}} - V_{i+1}). \quad (4.4)$$

Due to the continuity of flux and potential, equating the flux on each side of the

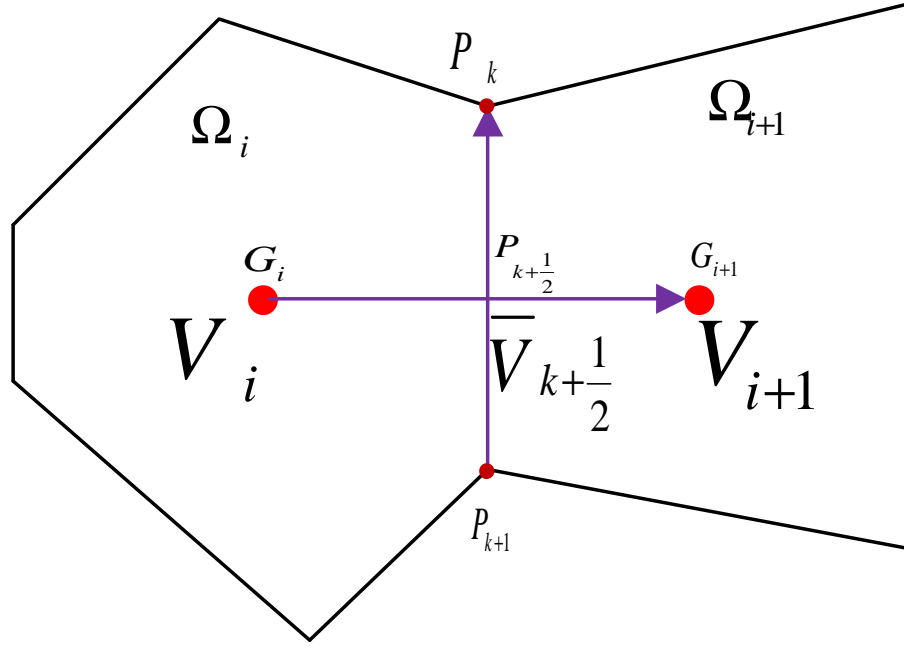


Figure 4.3: Two-point flux approximation definition with two cells Ω^i and Ω^{i+1} .

interface gives

$$F_i = F_{i+1}, \quad (4.5)$$

which can be used to solve $\bar{V}_{k+\frac{1}{2}}$

$$\bar{V}_{k+\frac{1}{2}} = \frac{T_i V_i + T_{i+1} V_{i+1}}{T_i + T_{i+1}}. \quad (4.6)$$

Inserting this expression back into Eqn (4.3) gives the flux expression

$$F_i = \frac{V_i - V_{i+1}}{\frac{1}{T_i} + \frac{1}{T_{i+1}}} = T_{k+\frac{1}{2}} (V_i - V_{i+1}). \quad (4.7)$$

Thus, the transmissibilities defined at the interface $T_{k+\frac{1}{2}}$ is the harmonic average

between the two neighboring cells.

$$\frac{1}{T_{k+\frac{1}{2}}} = \frac{1}{T_i} + \frac{1}{T_{i+1}}. \quad (4.8)$$

Since the face flux in this scheme only involves two neighboring cells, it is called Two-Point Flux Approximation (TPFA). It is a special case of the MPFA, where the off-diagonal entries in the matrix T of Eqn(4.16) become zero.

Both the rectangular mesh and the equilateral triangular mesh follows the TPFA scheme due to the orthogonality between centroids of adjacent volumes. For instance, the TPFA scheme will reduce to five-point scheme for rectangular mesh, which is equivalent to FDM. In addition, the popular Voronoi mesh which is the dual mesh of delaunay triangulation, has multiple flexible applications [49][50] with the TPFA scheme.

4.3.1.2 Mesh generation

In order to apply TPFA, unstructured PEBI or Voronoi grid was employed. The advantage of Voronoi grid is the edges that comprise a control volume are orthogonal to the segment connecting control volume centroids, which ensures TPFA is applicable to these meshes. As shown in Fig. 4.4, by making perpendicular bisectors of the black triangular mesh, the Voronoi cell marked with blue polygon is formed. Therefore, 2D Voronoi grid is considered as the dual mesh of the 2D delaunay triangular mesh. Thus, practically, Voronoi mesh generation follows the Delaunay triangulation procedure which has the property that no vertex lies inside any circumcircle of any triangle.

The challenge of the practical resistance extraction problem is how to generate a proper Voronoi grid to fit the non-orthogonal geometric interface or boundary. The main idea is to apply unstructured grids to the non-orthogonal domain only,

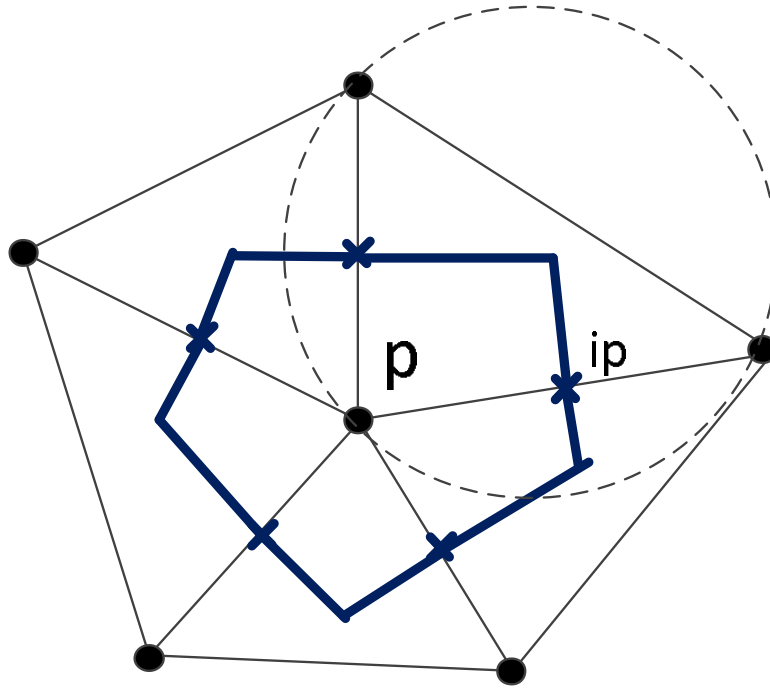


Figure 4.4: A Voronoi-Delaunay duality in two dimensions.

for instance, around the interface of two different conductivity. Because there exists complex geometries, for which flexible shapes of the unstructured grids come into playing a key role. For other domains such as the simulation background, we might use structured grids, like the rectangular mesh.

To generate mesh fitting the non-orthogonal interface, the domain around the interface which we name it as protection area has to be treated specifically. We pre-placed fixed pairs of nodes around the interface which will be the Voronoi cell centers along the interface. The corresponding Voronoi cell boundaries will yield the expected interface line segment as displayed in Fig. 4.5, where $d1$ is the fitting depth of the Voronoi cell center distance around the interface, which is set to be around $1/3$ of the rectangular grid size. While the Voronoi cell grid size $d2$ along the tangential

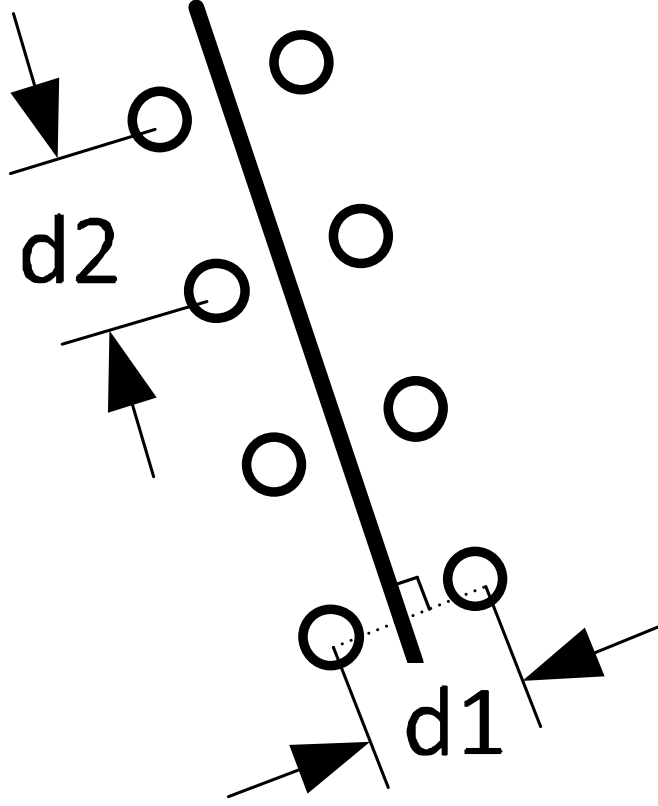


Figure 4.5: Assign fixed Voronoi cell centers to fit a non-orthogonal interface edge.

direction of the interface is usually chosen to be $1/3$ to $1/2$ of the rectangular grid size. Both of them are adjustable with the mesh density requirement. It is worth noting that since the geometries with non-orthogonal interfaces are usually convex, the approach we adopted to put fixed points can not guarantee the Voronoi mesh match the corner points exactly. However, the approximation can still ensure high accuracy for extraction due to the error from minor shifted interfaces is small. For the simulation background, we put the Cartesian grid points. Delaunay mesh points are composed of fixed points and Cartesian grid points after removing those with distance less than $\frac{d_1}{2}$ to the non-orthogonal interfaces. They will be the final Voronoi

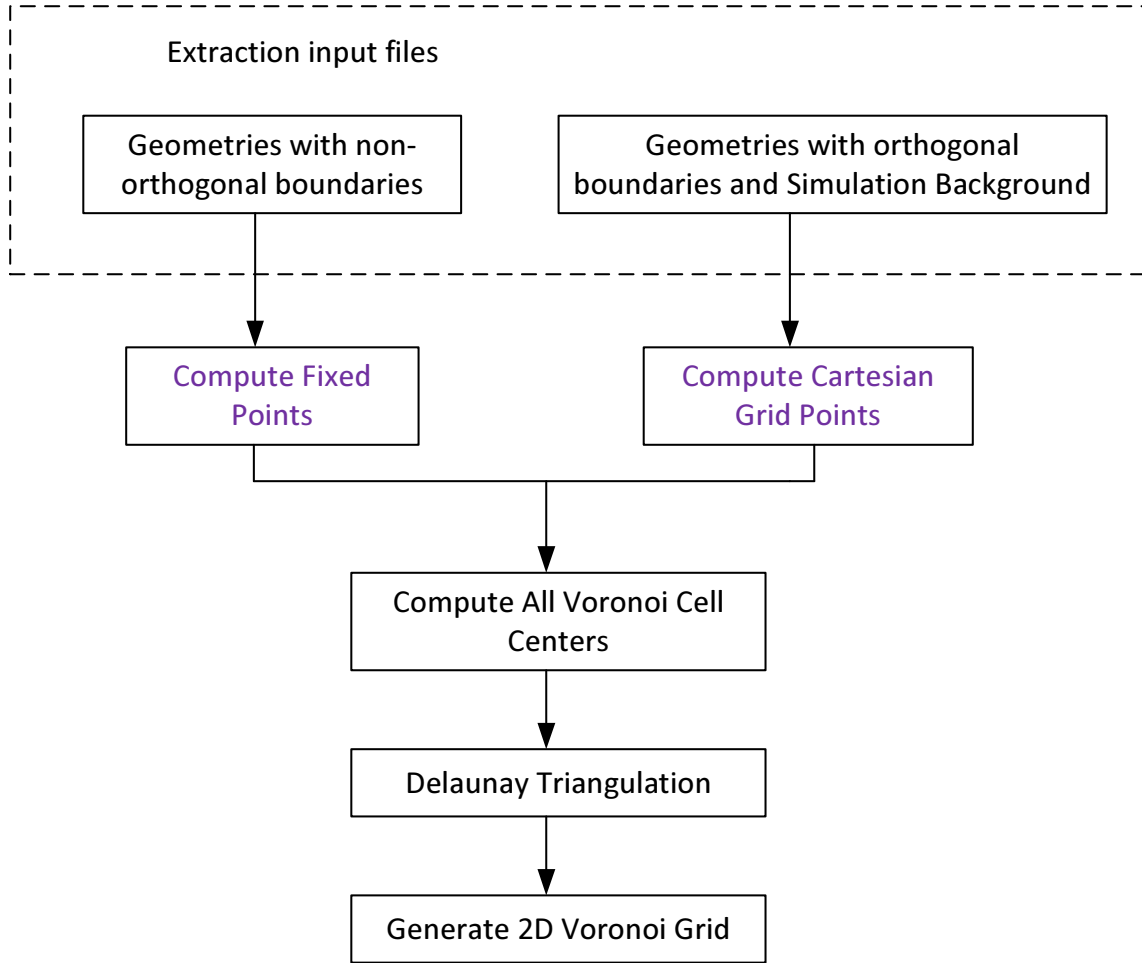


Figure 4.6: Mesh generation scheme to fit the non-orthogonal interfaces or boundaries.

cell centers. Fig. 4.6 summarizes the mesh generation flow.

4.3.1.3 Mesh Grid Optimization

It is necessary to improve the quality of the mesh since the accuracy of TPFA is highly related to it.

Multi-level Quad-Tree Local Grid Refinements (LGR) is a well known method for doing multi-level LGR for a structured grid in 2D. A rectangular or quadrilateral

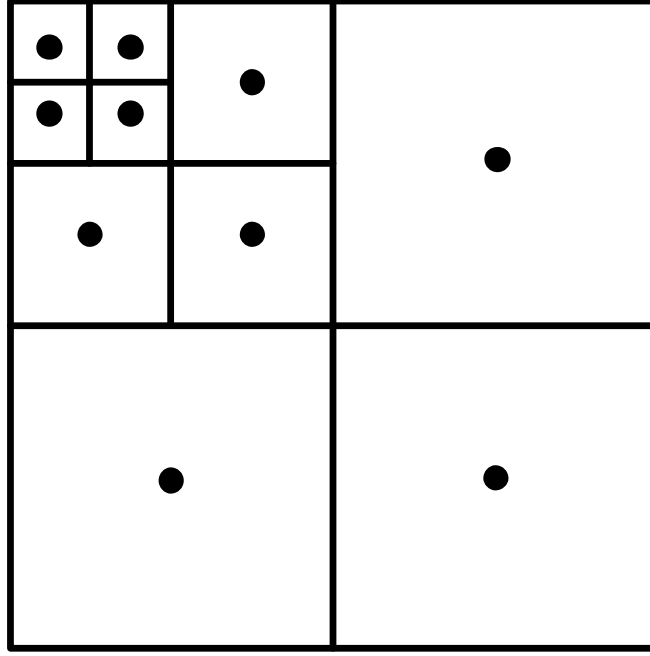


Figure 4.7: Multi-level quad-tree local grid refinement illustration.

grid cell is subdivided into four cells by connecting the middle points of the opposite edges. The four new grid points are the centers of the new cells which replace the original coarse cell grid point. Fig. 4.7 shows the basic idea of quad-tree LGR on a Cartesian grid. Multilevel quad-tree LGR method is used to generate successively denser grid points when an existing grid point is within a prescribed distance measure to the protection area of the non-orthogonal interfaces. The number of levels and distance measures can be controlled by the parameter settings. It can greatly improve the mesh density around the non-orthogonal interfaces, where the field changes fast, thus improve the extraction accuracy thereafter.

4.3.2 Multi-Point Flux Approximation

Suppose the computational mesh is composed of finite number of non-overlapping polygons. For any polygonal cell Ω_i , by integrating it over the control volume, and applying the divergence theorem, we obtain

$$\int_{\Omega_i} \nabla \cdot \sigma \nabla V dV = \oint_{\partial\Omega_i} \sigma \nabla V dA, \quad (4.9)$$

where $\partial\Omega_i$ is the face area of the cell Ω_i . The right hand side of Eqn (4.9) can be approximated by the sum of fluxes on all faces

$$\sum_f F_f = \sum_f (\sigma \nabla V)_f \cdot A_f, \quad (4.10)$$

where F_f is the diffusion flux on the face f .

4.3.2.1 Half Edge Flux Approximation in the Sub-cell

In this section, we will describe the numerical approach to approximate the half edge normal flux locally.

As demonstrated in Fig. 4.8, we denote $P_{k-\frac{1}{2}}$, $P_{k+\frac{1}{2}}$ the midpoint of the edges $[P_{k-1}, P_k]$ and $[P_k, P_{k+1}]$ separately. For any polygon Ω^i built with vertices P_j , $j = k-1, k, k+1, \dots$, sub-cell Ω_k^i is the quadrilateral constructed by connecting vertices $P_{k-\frac{1}{2}}$, P_k , $P_{k+\frac{1}{2}}$ and the centroid of cell Ω^i which is labeled as G_i . The inner angle between the edge $[P_{k-1}, P_k]$ and $[P_k, P_{k+1}]$ is θ_k^i . We assume the potential is defined on the edge $[P_{k-\frac{1}{2}}, P_k]$ and $[P_k, P_{k+\frac{1}{2}}]$ by two constant value $\bar{V}_{k-\frac{1}{2},k}$ and $\bar{V}_{k,k+\frac{1}{2}}$ respectively. And the half edge normal fluxes at edges $[P_{k-\frac{1}{2}}, P_k]$ and $[P_k, P_{k+\frac{1}{2}}]$ are denoted as $F_{k-\frac{1}{2},k}$, $F_{k,k+\frac{1}{2}}$. The direction of fluxes are defined counterclockwise with respect to the vertex P_k which is marked with arrow in Fig. 4.8.

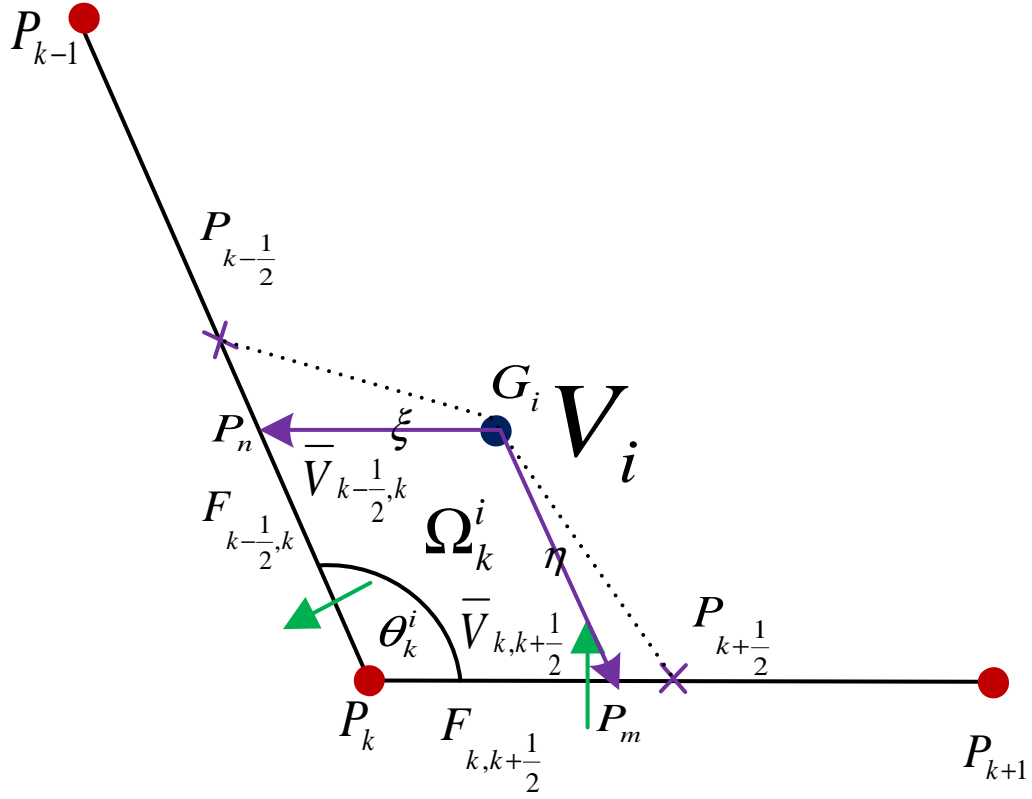


Figure 4.8: Sub-cell Ω_k^i constructed by vertices G_i , $P_{k-\frac{1}{2}}$, P_k and $P_{k+\frac{1}{2}}$.

In [47], Support Operators Method (SOM) has been applied to derive a local variational formulation within the sub-cell Ω_k^i ,

$$\begin{bmatrix} -F_{k-\frac{1}{2},k} \\ F_{k,k+\frac{1}{2}} \end{bmatrix} = \frac{\sigma_i}{4\omega_k^i} D \begin{bmatrix} 1 & -\cos\theta_k^i \\ -\cos\theta_k^i & 1 \end{bmatrix} D \begin{bmatrix} \bar{V}_{k-\frac{1}{2},k} - V_i \\ \bar{V}_{k,k+\frac{1}{2}} - V_i \end{bmatrix}, \quad (4.11)$$

where

$$D = \begin{bmatrix} |\overrightarrow{P_k P_{k-1}}| & 0 \\ 0 & |\overrightarrow{P_k P_{k+1}}| \end{bmatrix}, \quad (4.12)$$

σ_i is the conductivity coefficient in the cell Ω^i . ω_k^i is the corner volume, which is related to the vertex P_k satisfying

$$\sum_k \omega_k^i = A_{\Omega^i}, \quad (4.13)$$

where A_{Ω^i} is the area of the cell Ω^i . There are several possible choices for ω_k^i . In [51], the corner volumes of quadrilateral cells are

$$\omega_k^i = \frac{1}{4} \left| \overrightarrow{P_k P_{k-1}} \right| \left| \overrightarrow{P_k P_{k+1}} \right| \sin \theta_k^i. \quad (4.14)$$

Similarly, we can set it to be one third of the cell area for triangular cells,

$$\omega_k^i = \frac{1}{6} \left| \overrightarrow{P_k P_{k-1}} \right| \left| \overrightarrow{P_k P_{k+1}} \right| \sin \theta_k^i. \quad (4.15)$$

In such case, Eqn(4.11) can be simplified as

$$\begin{bmatrix} -F_{k-\frac{1}{2},k} \\ F_{k,k+\frac{1}{2}} \end{bmatrix} = T \begin{bmatrix} \bar{V}_{k-\frac{1}{2},k} - V_i \\ \bar{V}_{k,k+\frac{1}{2}} - V_i \end{bmatrix}, \quad (4.16)$$

where

$$T = \frac{3\sigma_i}{2 \cdot \overrightarrow{P_k P_{k+1}} \times \overrightarrow{P_k P_{k-1}}} \begin{bmatrix} \left| \overrightarrow{P_k P_{k-1}} \right|^2 & -\overrightarrow{P_k P_{k+1}} \cdot \overrightarrow{P_k P_{k-1}} \\ -\overrightarrow{P_k P_{k+1}} \cdot \overrightarrow{P_k P_{k-1}} & \left| \overrightarrow{P_k P_{k+1}} \right|^2 \end{bmatrix}. \quad (4.17)$$

The matrix T is symmetric. This matrix yields a linear relationship between the half edge normal fluxes and the half edge potential which will be used to generate the local diffusion matrix.

We could also derive the matrix T with the coordinate transformation, which is

consistent with the MPFA-C($\frac{1}{3}$) method [52] derived with the symmetric physical space or symmetric sub-cell transform space [48]. The parallelogram constructed by G_i and the edges $[P_{k-1}, P_k]$, $[P_k, P_{k+1}]$ have the intersection points P_m and P_n . We define a new coordinate system (ξ, η) as shown in Fig. 4.8.

From the chaining rule, we know that

$$\begin{bmatrix} V_\xi \\ V_\eta \end{bmatrix} = \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad (4.18)$$

which results in

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix}^{-1} \begin{bmatrix} V_\xi \\ V_\eta \end{bmatrix} = \frac{\begin{bmatrix} y_\eta & -y_\xi \\ -x_\eta & x_\xi \end{bmatrix}}{x_\xi y_\eta - x_\eta y_\xi} \begin{bmatrix} V_\xi \\ V_\eta \end{bmatrix}. \quad (4.19)$$

Hence

$$\begin{aligned} V_x &= \frac{V_\xi y_\eta - V_\eta y_\xi}{x_\xi y_\eta - x_\eta y_\xi}, \\ V_y &= \frac{-V_\xi x_\eta + V_\eta x_\xi}{x_\xi y_\eta - x_\eta y_\xi}. \end{aligned} \quad (4.20)$$

Suppose the coordinates for node G_i , P_n , P_m , P_k , P_{k+1} and P_{k-1} are (x_1, y_1) ,

(x_2, y_2) , (x_3, y_3) , (x_4, y_4) and (x_5, y_5) , (x_6, y_6) respectively. It is readily easy to get

$$\begin{aligned}
x_\xi &= \frac{x_2 - x_1}{\Delta\xi}, \\
y_\xi &= \frac{y_2 - y_1}{\Delta\xi}, \\
e_\xi &= \frac{(x_2 - x_1)\vec{i} + (y_2 - y_1)\vec{j}}{\Delta\xi}, \\
x_\eta &= \frac{x_3 - x_1}{\Delta\eta}, \\
y_\eta &= \frac{y_3 - y_1}{\Delta\eta}, \\
e_\eta &= \frac{(x_3 - x_1)\vec{i} + (y_3 - y_1)\vec{j}}{\Delta\eta}, \\
L_x &= y_2 - y_1, \\
L_y &= -(x_2 - x_1), \\
R_x &= y_3 - y_1, \\
R_y &= -(x_3 - x_1),
\end{aligned}$$

where $\vec{L} = L_x\vec{e}_x + L_y\vec{e}_y$ and $\vec{R} = R_x\vec{e}_x + R_y\vec{e}_y$ are the area vectors on the edges $[P_k, P_m]$ and $[P_n, P_k]$ respectively.

The denominator in Eqn (4.20) can be rewritten as

$$x_\xi y_\eta - x_\eta y_\xi = \frac{x_2 - x_1}{\Delta\xi} \frac{y_3 - y_1}{\Delta\eta} - \frac{x_3 - x_1}{\Delta\eta} \frac{y_2 - y_1}{\Delta\xi}.$$

Thus, the fluxes at the edges $[P_k, P_m]$ and $[P_k, P_n]$ satisfy

$$\begin{aligned}
F_{k,m} &= \sigma_i \nabla V \cdot \vec{L} \\
&= \sigma_i V_x L_x + V_y L_y \\
&= \sigma_i \frac{V_\xi y_\eta - V_\eta y_\xi}{x_\xi y_\eta - x_\eta y_\xi} L_x + \frac{-V_\xi x_\eta + V_\eta x_\xi}{x_\xi y_\eta - x_\eta y_\xi} L_y \\
&= \sigma_i \left(\frac{L_x y_\eta - L_y x_\eta}{x_\xi y_\eta - x_\eta y_\xi} \right) V_\xi + \left(\frac{-L_x y_\xi + L_y x_\xi}{x_\xi y_\eta - x_\eta y_\xi} \right) V_\eta \\
&= \sigma_i \frac{(y_2 - y_1)(y_3 - y_1) + (x_2 - x_1)(x_3 - x_1)}{((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))/\Delta\xi} V_\xi \\
&\quad + \sigma_i \frac{-(y_2 - y_1)^2 - (x_2 - x_1)^2}{((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))/\Delta\eta} V_\eta \\
&= \sigma_i \frac{(y_2 - y_1)(y_3 - y_1) + (x_2 - x_1)(x_3 - x_1)}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)} (V_{P_n} - V_i) \\
&\quad + \sigma_i \frac{-(y_2 - y_1)^2 - (x_2 - x_1)^2}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)} (V_{P_m} - V_i),
\end{aligned}$$

and

$$\begin{aligned}
F_{k,n} &= \sigma_i \nabla V \cdot \vec{R} \\
&= \sigma_i V_x R_x + V_y R_y \\
&= \sigma_i \frac{V_\xi y_\eta - V_\eta y_\xi}{x_\xi y_\eta - x_\eta y_\xi} R_x + \frac{-V_\xi x_\eta + V_\eta x_\xi}{x_\xi y_\eta - x_\eta y_\xi} R_y \\
&= \sigma_i \left(\frac{R_x y_\eta - R_y x_\eta}{x_\xi y_\eta - x_\eta y_\xi} \right) V_\xi + \left(\frac{-R_x y_\xi + R_y x_\xi}{x_\xi y_\eta - x_\eta y_\xi} \right) V_\eta \\
&= \sigma_i \frac{(y_3 - y_1)^2 + (x_3 - x_1)^2}{((x_1 - x_2)(y_1 - y_3) - (x_1 - x_3)(y_1 - y_2))/\Delta\xi} V_\xi \\
&\quad + \sigma_i \frac{-(y_2 - y_1)(y_3 - y_1) - (x_2 - x_1)(x_3 - x_1)}{((x_1 - x_2)(y_1 - y_3) - (x_1 - x_3)(y_1 - y_2))/\Delta\eta} V_\eta \\
&= \sigma_i \frac{(y_3 - y_1)^2 + (x_3 - x_1)^2}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)} (V_{P_n} - V_i) \\
&\quad + \sigma_i \frac{-(y_2 - y_1)(y_3 - y_1) - (x_2 - x_1)(x_3 - x_1)}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)} (V_{P_m} - V_i).
\end{aligned}$$

They can be summarized in the compact form

$$\begin{bmatrix} -F_{k,n} \\ F_{k,m} \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} V_{P_n} - V_i \\ V_{P_m} - V_i \end{bmatrix},$$

where

$$\begin{aligned} T_{11} &= \sigma_i \frac{-(y_3 - y_1)^2 - (x_3 - x_1)^2}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}, \\ T_{12} = T_{21} &= \sigma_i \frac{(y_2 - y_1)(y_3 - y_1) + (x_2 - x_1)(x_3 - x_1)}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}, \\ T_{22} &= \sigma_i \frac{-(y_2 - y_1)^2 - (x_2 - x_1)^2}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}. \end{aligned}$$

It is obvious to find that

$$\begin{aligned} (x_1, y_1) &= \frac{1}{3}((x_4, y_4) + (x_5, y_5) + (x_6, y_6)), \\ (x_2, y_2) &= \frac{1}{3}(x_6, y_6) + \frac{2}{3}(x_4, y_4), \\ (x_3, y_3) &= \frac{1}{3}(x_5, y_5) + \frac{2}{3}(x_4, y_4), \\ F_{k-\frac{1}{2},k} &= \frac{3}{2}F_{k,n}, \\ F_{k,k+\frac{1}{2}} &= \frac{3}{2}F_{k,m}, \end{aligned}$$

which can be used to derive the relationship between the half edge fluxes and the half edge potential

$$\begin{bmatrix} -F_{k-\frac{1}{2},k} \\ F_{k,k+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} T'_{11} & T'_{12} \\ T'_{21} & T'_{22} \end{bmatrix} \begin{bmatrix} \bar{V}_{k-\frac{1}{2},k} - V_i \\ \bar{V}_{k,k+\frac{1}{2}} - V_i \end{bmatrix},$$

where

$$\begin{aligned}
T'_{11} &= \frac{3\sigma_i}{2} \frac{-(y_4 - y_6)^2 - (x_4 - x_6)^2}{(x_4 - x_5)(y_4 - y_6) - (x_4 - x_6)(y_4 - y_5)}, \\
T'_{12} = T'_{21} &= \frac{3\sigma_i}{2} \frac{(y_4 - y_6)(y_4 - y_5) + (x_4 - x_6)(x_4 - x_5)}{(x_4 - x_5)(y_4 - y_6) - (x_4 - x_6)(y_4 - y_5)}, \\
T'_{22} &= \frac{3\sigma_i}{2} \frac{-(y_4 - y_5)^2 - (x_4 - x_5)^2}{(x_4 - x_5)(y_4 - y_6) - (x_4 - x_6)(y_4 - y_5)}. \tag{4.21}
\end{aligned}$$

They are consistent with the Eqn (4.17) except the sign which depends on the reference flux direction and will not affect the result.

4.3.2.2 Discretization of the Local Diffusion Matrix in the Vertex

Suppose there is an internal node P_k surrounded by cells Ω^j with constant potential V_j at the centroid, where $j = i + 1, i + 2, \dots, i + r$. We label the adjacent nodes as $P_{k+1}, P_{k+2}, \dots, P_{k+r}$. We also denote by P_a the midpoint of the edge $[P_k, P_{k+1}]$. The average potential and normal face flux on the edge $[P_k, P_a]$ are defined by $\bar{V}_{k,a}$ and $F_{k,a}$ respectively. Similar definition for the edges between P_k and other adjacent nodes. As shown in Fig. 4.9, without loss of generality, we set the number of cells $r = 4$ for the following derivation.

In the cell passing through the vertices P_k, P_{k+1} and P_{k+2} , based on the local half edge flux Eqn (4.16), the flux normal to the edge $[P_k, P_a]$ and $[P_k, P_b]$ satisfy

$$\begin{bmatrix} -F_{k,b} \\ F_{k,a} \end{bmatrix} = T^{i+1} \begin{bmatrix} \bar{V}_{k,b} - V_{i+1} \\ \bar{V}_{k,a} - V_{i+1} \end{bmatrix},$$

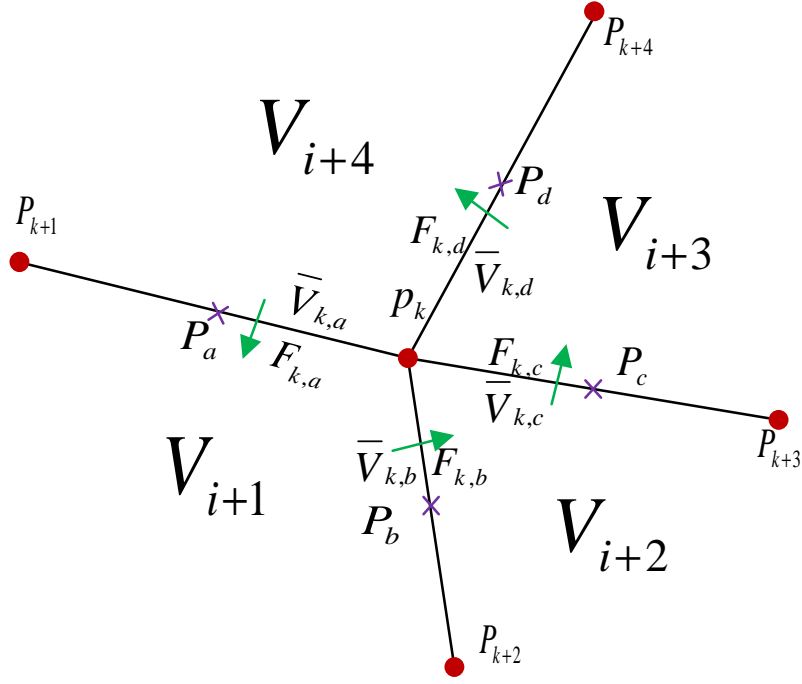


Figure 4.9: Notation around an internal vertex p_k .

where T^{i+1} is the local matrix in the sub-cell Ω_k^{i+1} . It is easy to obtain

$$\begin{bmatrix} \bar{V}_{k,b} - V_{i+1} \\ \bar{V}_{k,a} - V_{i+1} \end{bmatrix} = \tilde{T}^{i+1} \begin{bmatrix} -F_{k,b} \\ F_{k,a} \end{bmatrix},$$

where \tilde{T}^{i+1} is the inverse matrix of T^{i+1} which is also symmetric. We have similar

formula for fluxes in other cells:

$$\begin{aligned} \begin{bmatrix} \bar{V}_{k,c} - V_{i+2} \\ \bar{V}_{k,b} - V_{i+2} \end{bmatrix} &= \tilde{T}^{i+2} \begin{bmatrix} -F_{k,c} \\ F_{k,b} \end{bmatrix}, \\ \begin{bmatrix} \bar{V}_{k,d} - V_{i+3} \\ \bar{V}_{k,c} - V_{i+3} \end{bmatrix} &= \tilde{T}^{i+3} \begin{bmatrix} -F_{k,d} \\ F_{k,c} \end{bmatrix}, \\ \begin{bmatrix} \bar{V}_{k,a} - V_{i+4} \\ \bar{V}_{k,d} - V_{i+4} \end{bmatrix} &= \tilde{T}^{i+4} \begin{bmatrix} -F_{k,a} \\ F_{k,d} \end{bmatrix}. \end{aligned}$$

We can solve these linear equations, and rewrite them in the following form

$$\begin{bmatrix} F_{k,b} - F_{k,a} \\ F_{k,c} - F_{k,b} \\ F_{k,d} - F_{k,c} \\ F_{k,a} - F_{k,d} \end{bmatrix} = M \begin{bmatrix} V_{i+1} \\ V_{i+2} \\ V_{i+3} \\ V_{i+4} \end{bmatrix}, \quad (4.22)$$

where

$$M = L^T S^{-1} L, \quad (4.23)$$

$$L = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad (4.24)$$

and

$$S = \begin{bmatrix} \tilde{T}_{22}^{i+1} + \tilde{T}_{11}^{i+4} & -\tilde{T}_{21}^{i+1} & 0 & -\tilde{T}_{12}^{i+4} \\ -\tilde{T}_{12}^{i+1} & \tilde{T}_{11}^{i+1} + \tilde{T}_{22}^{i+2} & -\tilde{T}_{21}^{i+2} & 0 \\ 0 & -\tilde{T}_{12}^{i+2} & \tilde{T}_{11}^{i+2} + \tilde{T}_{22}^{i+3} & -\tilde{T}_{21}^{i+3} \\ -\tilde{T}_{21}^{i+4} & 0 & -\tilde{T}_{12}^{i+3} & \tilde{T}_{11}^{i+3} + \tilde{T}_{22}^{i+4} \end{bmatrix}. \quad (4.25)$$

L is the linear combination of an identity matrix and a permutation matrix. It is clear that S is a symmetric matrix, which results in a symmetric local diffusion matrix M .

The expression $F_{k,b} - F_{k,a}$ represents the outward diffusion flux in the cell Ω^{i+1} contributed by node P_k . Therefore, Eqn (4.22) yields a linear relation between the outward half face fluxes of all adjacent cells and the corresponding cell-centered potentials. If we define $[M_G]$ as an assemble of the local diffusion matrix, it has been proven that it is SPD in [48].

4.3.2.3 Boundary Conditions

There are two type of boundary conditions: Neumann and Dirichlet. In this section, we will analyze their contribution to the diffusion matrix.

4.3.2.3.1 Neumann Boundary Condition Suppose P_k is a boundary vertex, and the edges $[P_k, P_{k+1}]$ and $[P_k, P_{k+4}]$ are located on the Neumann boundary. Then the boundary flux will satisfy

$$F_{k,a} = F_{k,d} = 0. \quad (4.26)$$

By inserting these conditions to Eqn(4.22), the unknown face fluxes can be expressed as

$$\begin{bmatrix} F_{k,b} - F_{k,a} \\ F_{k,c} - F_{k,b} \\ F_{k,d} - F_{k,c} \end{bmatrix} = M_N \begin{bmatrix} V_{i+1} \\ V_{i+2} \\ V_{i+3} \end{bmatrix}, \quad (4.27)$$

where

$$M_N = L_N^T S_N^{-1} L_N,$$

$$L_N = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix},$$

and

$$S_N = \begin{bmatrix} \tilde{T}_{11}^{i+1} + \tilde{T}_{22}^{i+2} & -\tilde{T}_{21}^{i+2} \\ -\tilde{T}_{12}^{i+2} & \tilde{T}_{11}^{i+2} + \tilde{T}_{22}^{i+3} \end{bmatrix}.$$

The local diffusion matrix M_N is also symmetric, and matrix L_N and S_N are the submatrix of L and S individually, where the rule to remove rows and columns is easy to recognize.

4.3.2.3.2 Dirichlet Boundary Condition Assume vertex P_k is located on the Dirichlet boundary with two boundary edges $[P_k, P_{k+1}]$ and $[P_k, P_{k+4}]$. If both two edges are on the Dirichlet Boundary with constant potential value W

$$\bar{V}_{k+a} = \bar{V}_{k+d} = W. \quad (4.28)$$

Then the face fluxes of the cells impinging on the node P_k can be written as

$$\begin{bmatrix} F_{k,b} - F_{k,a} \\ F_{k,c} - F_{k,b} \\ F_{k,d} - F_{k,c} \end{bmatrix} = M_D \begin{bmatrix} V_{i+1} \\ V_{i+2} \\ V_{i+3} \end{bmatrix} + C_D W, \quad (4.29)$$

where

$$M_D = L_D^T S_D^{-1} L_D,$$

$$L_D = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix},$$

and

$$S_D = \begin{bmatrix} \tilde{T}_{22}^{i+1} & -\tilde{T}_{21}^{i+1} & 0 & 0 \\ -\tilde{T}_{12}^{i+1} & \tilde{T}_{11}^{i+1} + \tilde{T}_{22}^{i+2} & -\tilde{T}_{21}^{i+2} & 0 \\ 0 & -\tilde{T}_{12}^{i+2} & \tilde{T}_{11}^{i+2} + \tilde{T}_{22}^{i+3} & -\tilde{T}_{21}^{i+3} \\ 0 & 0 & -\tilde{T}_{12}^{i+3} & \tilde{T}_{11}^{i+3} \end{bmatrix},$$

as well as

$$C_D = L_D^T S_D^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}.$$

It is easy to obtain the total flux along the Dirichlet boundary for the vertex P_k through $F_{k,d} - F_{k,a}$ which is the sum of matrix columns for Eqn(4.29).

If the vertex P_k is on the corner, then one of the boundary edge will be located on the Neumann boundary. For instance,

$$F_{k,a} = 0, \quad \bar{V}_{k+d} = W. \quad (4.30)$$

In such case, the face fluxes of the cells surrounding the node P_k can be written as

$$\begin{bmatrix} F_{k,b} - F_{k,a} \\ F_{k,c} - F_{k,b} \\ F_{k,d} - F_{k,c} \end{bmatrix} = M_{ND} \begin{bmatrix} V_{i+1} \\ V_{i+2} \\ V_{i+3} \end{bmatrix} + C_{ND}W, \quad (4.31)$$

where

$$M_{ND} = L_{ND}^T S_{ND}^{-1} L_{ND},$$

$$L_{ND} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix},$$

and

$$S_{ND} = \begin{bmatrix} \tilde{T}_{11}^{i+1} + \tilde{T}_{22}^{i+2} & -\tilde{T}_{21}^{i+2} & 0 \\ -\tilde{T}_{12}^{i+2} & \tilde{T}_{11}^{i+2} + \tilde{T}_{22}^{i+3} & -\tilde{T}_{21}^{i+3} \\ 0 & -\tilde{T}_{12}^{i+3} & \tilde{T}_{11}^{i+3} \end{bmatrix},$$

as well as

$$C_{ND} = L_{ND}^T S_{ND}^{-1} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}.$$

4.4 Finite Element Method

Finite element method (FEM) is a numerical method like the finite difference method but is more general and powerful in its application to real-world problems that involve complicated physics, geometry and boundary conditions.

4.4.1 Traditional Finite Element Method

In FEM, a given domain is viewed as a collection of sub-domains, and over each domain the governing equation is approximated by any of the traditional variational methods. In the classic Galerkin method, the weak form is a variational statement of the problem in which we integrate against a test function. This has the effect of relaxing the problem: instead of finding an exact solution everywhere, we are finding a solution that satisfies the strong form on average over the domain. From other

point of view, it weakens the original differential equation to integral or sum equation which is much more easy to solve.

4.4.1.1 Weak Formulation and Global Stiffness Matrix

Starting with the strong form

$$-\nabla \cdot \sigma \nabla V = 0, \quad (4.32)$$

by multiplying a test function ψ

$$\begin{aligned} -\int_{\Omega} \psi \nabla \cdot \sigma \nabla V &= 0 \\ -\left(\int_{\partial\Omega} \psi \cdot n \cdot \sigma \nabla V - \int_{\Omega} \nabla \psi \cdot \sigma \nabla V \right) &= 0 \\ \int_{\Omega} \nabla \psi \cdot \sigma \nabla V &= 0, \end{aligned}$$

it has been turned to the weak form

$$(\nabla \psi, \sigma \nabla V) = 0, \quad \forall \psi \in H^1. \quad (4.33)$$

Since the solution V comes from an infinite dimensional function space, it is impossible to handle objects with infinitely many coefficients, we seek a finite dimensional trial function to approximate it instead,

$$V_h = \sum_{j=1}^N U_j \psi_j(x), \quad (4.34)$$

where $\psi_j(x)$ are the finite element shape function, U_j are unknown expansion coefficients, the total number N represents degree of freedom. To determine the N

coefficients, test with N basis functions

$$(\nabla\psi_i, \sigma\nabla V_h) = 0, \quad \forall i = 1, \dots, N.$$

If basis function are linearly independent, this yields N equations for N coefficients.

$$\sum_{j=1}^N (\nabla\psi_i, \sigma\nabla\psi_j) U_j = 0, \quad \forall i = 1, \dots, N,$$

which can be written in the compact linear system

$$[\mathbf{K}]U = 0,$$

where $[\mathbf{K}]$ is the global stiffness matrix defined by $\mathbf{K}_{ij} = (\nabla\psi_i, \sigma\nabla\psi_j) = \sum_e \sigma_e \int_e \nabla\psi_i \nabla\psi_j$.

It has unique solution with the well-defined Dirichlet boundary conditions.

4.4.1.2 Linear Basis Function and Local Stiffness Matrix

Suppose the computational domain is composed of triangular elements. If the potential distribution within a single triangular cell follows

$$V_h(x, y) = a_0 + a_1x + a_2y, \tag{4.35}$$

which is linear in both x and y directions. $V_h(x, y)$ is uniquely defined on a triangle element by the values of V_h at the vertices of the triangle (x_1, y_1) , (x_2, y_2) , (x_3, y_3) ,

$$V_1 \equiv V_h(x_1, y_1) = a_0 + a_1x_1 + a_2y_1,$$

$$V_2 \equiv V_h(x_2, y_2) = a_0 + a_1x_2 + a_2y_2,$$

$$V_3 \equiv V_h(x_3, y_3) = a_0 + a_1x_3 + a_2y_3,$$

which can be rewritten as

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \equiv M \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}.$$

The coefficient can be solved

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = M^{-1} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \frac{1}{|M|} \begin{bmatrix} x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_2 - x_2y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}.$$

If we introduce the parameters α , β and γ

$$\alpha_i = x_jy_k - x_ky_j,$$

$$\beta_i = y_j - y_k,$$

$$\gamma_i = -(x_j - x_k),$$

where $i \neq j \neq k$ and they permute in a natural order, Eqn (4.35) can be written as

$$\begin{aligned} V_h(x, y) &= \frac{1}{|M|} [(\alpha_1V_1 + \alpha_2V_2 + \alpha_3V_3) + (\beta_1V_1 + \beta_2V_2 + \beta_3V_3)x + (\gamma_1V_1 + \gamma_2V_2 + \gamma_3V_3)y] \\ &= \frac{1}{|M|} [(\alpha_1 + \beta_1x + \gamma_1y)V_1 + (\alpha_2 + \beta_2x + \gamma_2y)V_2 + (\alpha_3 + \beta_3x + \gamma_3y)V_3] \\ &= \sum_{i=1}^3 V_i \psi_i(x, y). \end{aligned} \tag{4.36}$$

Comparing with Eqn (4.34), the basis function or linear interpolation function at

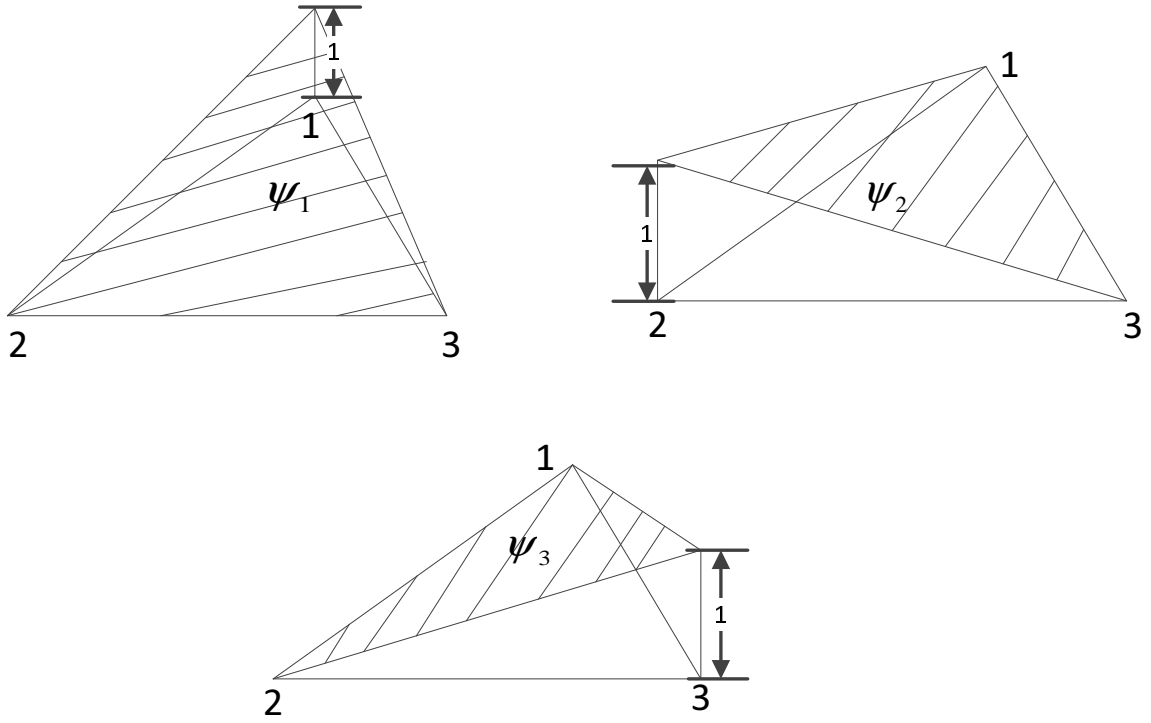


Figure 4.10: Linear interpolation functions for triangle element.

the node i is

$$\psi_i(x, y) = \frac{1}{|M|}(\alpha_i + \beta_i x + \gamma_i y), \quad (4.37)$$

with the properties

$$\begin{aligned} \psi_i(x_j, y_j) &= \delta_{i,j}, \\ \sum_{i=1}^3 \psi_i &= 1. \end{aligned}$$

The basis function at each vertex is demonstrated in Fig. 4.10.

The basis functions are used to build the local stiffness matrix \mathbf{K}^e which is defined

as $\mathbf{K}_{ij}^e = \int_e \sigma_e \nabla \psi_i \nabla \psi_j$ within each individual triangular cell. From

$$\begin{aligned}\frac{\partial \psi_i}{\partial x} &= \frac{\beta_i}{|M|}, \\ \frac{\partial \psi_i}{\partial y} &= \frac{\gamma_i}{|M|},\end{aligned}$$

we obtain

$$\int_e \sigma_e \nabla \psi_i \nabla \psi_j = \sigma_e \frac{\beta_i \beta_j + \gamma_i \gamma_j}{|M|^2} \frac{|M|}{2} \quad (4.38)$$

which is the contribution to the entry $\mathbf{K}_{i,j}$ from the e th triangular cell.

4.4.2 Immersed Finite Element Method

In the immersed finite element method (IFEM) [53][54], the basis functions are constructed to satisfy the jump conditions across the interface. The advantage of this method is that it can be considered as an extended finite difference method when cartesian triangulation is utilized to generate the mesh.

4.4.2.1 From FEM to IFEM

It is easy to verify that FEM with Cartesian triangular mesh are equivalent to FDM with Cartesian grid as displayed in Fig. 4.11.

We expect the Cartesian grid being applied to the computational domain where there is a geometry with non-orthogonal boundary as shown in Fig. 4.12(a). In Fig. 4.12(b), the mesh is replaced with Cartesian triangulation. We can divide all the cells into two groups, the non-interface elements and the interface elements. If there is no interface or boundary line passing through the interior of the element, we call it a non-interface element, the linear basis shape functions are equivalent to FDM applied on Cartesian grid. Otherwise, the element is an interface element

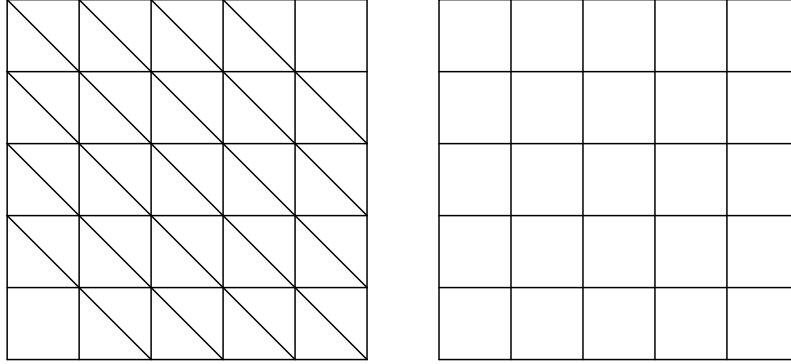


Figure 4.11: Cartesian triangulation grid and Cartesian grid.

which are split into several parts by the interface lines, we need to take the interface conditions into consideration for basis function construction. As the demonstration in Fig. 4.12(c), IFEM is composed of applying FDM to non-interface parts, and extending FEM to interface right triangle elements which is marked with red color. The method to extend FEM for triangle element with interface lines will be described in the following parts.

4.4.2.2 Non-conforming Basis Function for Interface Element

Suppose the computational domain Ω is separated by an interface Γ into two parts Ω^+ and Ω^- , and the conductivity coefficient σ has two pieces separated by the interface Γ

$$\sigma(x, y) = \begin{cases} \sigma^+ & (x, y) \in \Omega^+ \\ \sigma^- & (x, y) \in \Omega^- \end{cases} \quad (4.39)$$

For instance, the interface triangle cell $\triangle ABC$ as shown in Fig. 4.13 is separated into two parts with conductivity σ^+ in triangle $\triangle ADE$ and σ^- in quadrilateral

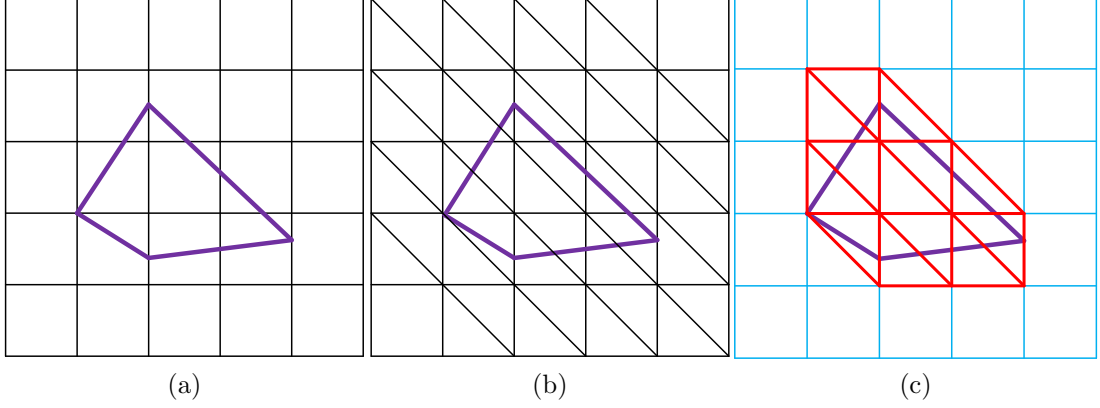


Figure 4.12: From FEM to IFEM: (a) geometry with non-orthogonal boundary in Cartesian grid; (b) geometry with non-orthogonal boundary in Cartesian triangular grid; (c) Cartesian grid for non-interface element and Cartesian triangular grid for interface element.

$\square DBCE$. Suppose the potential is piecewise linear in $\triangle ABC$ cell as follows:

$$\begin{aligned}
 V^+(x, y) &= a_0 + a_1x + a_2y, & (x, y) \in \triangle ABC \\
 V^-(x, y) &= b_0 + b_1x + b_2y. & (x, y) \in \square DBCE
 \end{aligned} \tag{4.40}$$

The potential continuity condition at node D and node E are

$$\begin{aligned}
 V^+(D) &= V^-(D), \\
 V^+(E) &= V^-(E).
 \end{aligned} \tag{4.41}$$

In addition, the normal flux at the interface should be conservative

$$\sigma^+ \frac{\partial V^+}{\partial n} = \sigma^- \frac{\partial V^-}{\partial n}. \tag{4.42}$$

Suppose the slope of the interface line segment within $\triangle ABC$ is α , then the normal

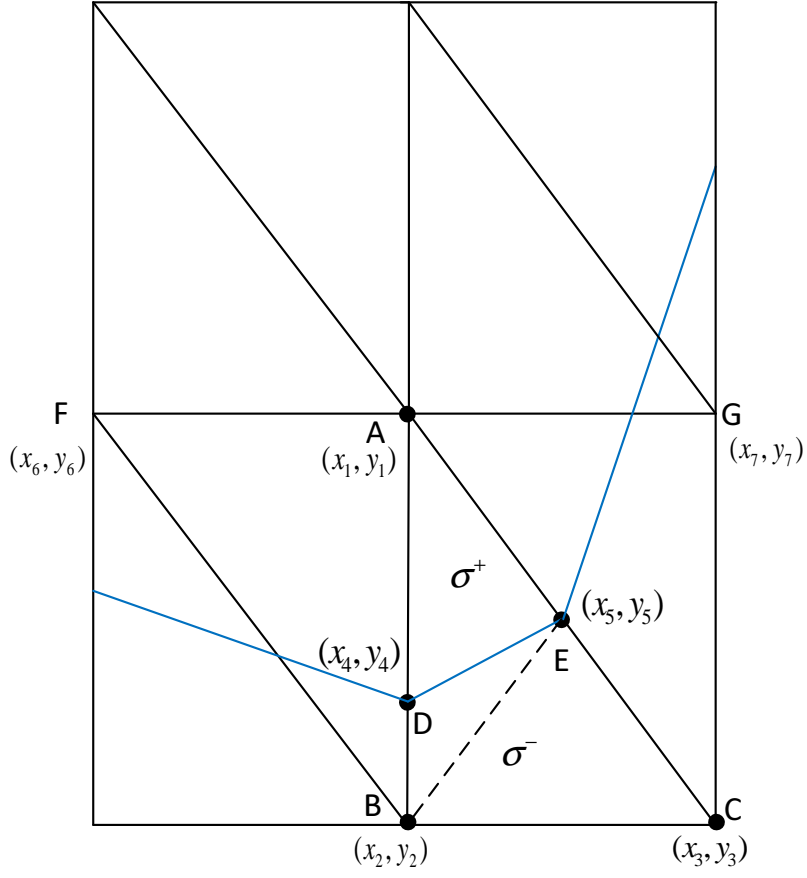


Figure 4.13: Cartesian triangles with an interface line cutting through.

direction vector of the line segment is $(\alpha, -1)$. Since

$$\frac{\partial V}{\partial n} = \nabla V \cdot \vec{n},$$

Eqn (4.42) can be rewritten as

$$\sigma^+(a_1\alpha - a_2) = \sigma^-(b_1\alpha - b_2). \quad (4.43)$$

Combining with potential equation in terms of node A , B and C , the new local linear

system becomes

$$\begin{bmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_2 & y_2 \\ 0 & 0 & 0 & 1 & x_3 & y_3 \\ 1 & x_4 & y_4 & -1 & -x_4 & -y_4 \\ 1 & x_5 & y_5 & -1 & -x_5 & -y_5 \\ 0 & \alpha & -1 & 0 & -\frac{\sigma^-}{\sigma^+}\alpha & \frac{\sigma^-}{\sigma^+} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix} \equiv M \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} V(A) \\ V(B) \\ V(C) \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (4.44)$$

We can obtain two separate linear shape functions coefficients

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = W^{\sigma^+} \begin{bmatrix} V(A) \\ V(B) \\ V(C) \end{bmatrix},$$

and

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = W^{\sigma^-} \begin{bmatrix} V(A) \\ V(B) \\ V(C) \end{bmatrix},$$

after solving this linear system. Fig .4.14 shows the modified basis function at node A with IFEM, where the changes at the interface line are continuous. It is easy to recognize the small discontinuous jumps along the element boundaries, thus the basis function is non-conforming.

The contribution term to the global entry $\mathbf{K}_{i,j}$ is also separated into two parts:

$$\int_{\triangle ABC} \sigma_{\triangle ABC} \nabla \psi_i \nabla \psi_j = \int_{\triangle ADE} \sigma^+ \nabla \psi_i \nabla \psi_j + \int_{\square DBCE} \sigma^- \nabla \psi_i \nabla \psi_j.$$

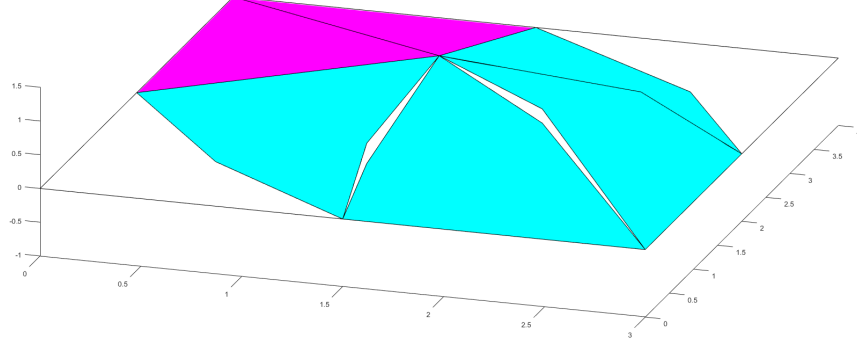


Figure 4.14: The basis function at node A in Fig.4.13 with IFEM when $\frac{\sigma^+}{\sigma^-} = 10$.

The gradient calculation for each part is the same as the traditional approach.

4.4.2.3 Conforming Basis Function for Interface Element

Compared with non-conforming IFEM, the conforming IFEM manually force the continuity at the element boundaries. For instance, in the $\triangle ABC$, we need to modify the non-conforming basis functions so that the local basis functions can be not only continuous at the edge DE , but also be continuous along the edge AB and AC . After obtaining linear basis function in $\triangle ABC$, the value at node D will be

$$V(D)_{\triangle ABC} = [1, x_4, y_4] \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}_{\triangle ABC} \equiv O_{\triangle ABC}^D \begin{bmatrix} V(A) \\ V(B) \\ V(C) \end{bmatrix}.$$

Similar relationship in $\triangle AFB$

$$V(D)_{\triangle AFB} = O_{\triangle AFB}^D \begin{bmatrix} V(A) \\ V(F) \\ V(B) \end{bmatrix}.$$

We choose the average value of D as the continuity node along the edge AB

$$\overline{V(D)} = \frac{1}{2}(V(D)_{\triangle ABC} + V(D)_{\triangle AFB}) \equiv O^D \begin{bmatrix} V(A) \\ V(B) \\ V(C) \\ V(F) \end{bmatrix}.$$

Similarly, the value at node E is adjusted to be the average of values with non-conforming basis functions at triangle $\triangle ABC$ and $\triangle ACG$,

$$\overline{V(E)} \equiv O^E \begin{bmatrix} V(A) \\ V(B) \\ V(C) \\ V(G) \end{bmatrix}.$$

To calculate the basis function for $\triangle ABC$ after boundary node adjustment, we need to split the triangle into three parts, $\triangle ADE, \triangle DBE, \triangle EBC$ or $\triangle ADE, \triangle DBC, \triangle DCE$. The sub-triangles need to satisfy the requirement that the maximum acute angle is greater than or equal to $\pi/4$ or the supplementary angle of the obtuse angle is not less than $\pi/4$.

For each sub-triangle, the basis function is piecewise linear function determined by the values at the corresponding vertices. For example, the basis function at

$\triangle ADE$ needs to satisfy

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_4 & y_4 \\ 1 & x_5 & y_5 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}_{\triangle ADE} = \begin{bmatrix} E_1 \\ O^D \\ O^E \end{bmatrix}_{5 \times 5} \cdot \begin{bmatrix} V(A) \\ V(B) \\ V(C) \\ V(F) \\ V(G) \end{bmatrix},$$

where E_1 is standard unit vector of length 5.

From the above analysis, we can find that both the non-conforming and conforming IFEM will result in a nine point stencil, with the same degree of freedom as the original cartesian triangulation. When the conductivity ratio is 1, there is no interface, the stencil will reduce to the original five point FDM stencil. It has been proved that the accuracy of conforming IFEM is second order accurate in terms of V , while the accuracy of non-conforming IFEM is between first order and second order in [53]. However, since the difference in the following experiments is trivial, we only show the simulation results with non-conforming IFEM.

4.5 Simulation and Evaluations

4.5.1 Resistance Extraction of non-rectilinear Conductor

There is a simple resistance extraction case where there is a very thin diagonal stripe shaped conductor. All dimensional detail are shown in Fig. 4.15, the non-rectilinear stripe shaped conductor can be any rotation angle α . We use FEM solver to obtain the reference results, when $\alpha = 30^\circ$, $R = 3.34\Omega$ with $dof = 19696$, when $\alpha = 45^\circ$, $R = 3.22\Omega$ with $dof = 19664$, while when $\alpha = 60^\circ$, $R = 3.06\Omega$ with $dof = 19740$.

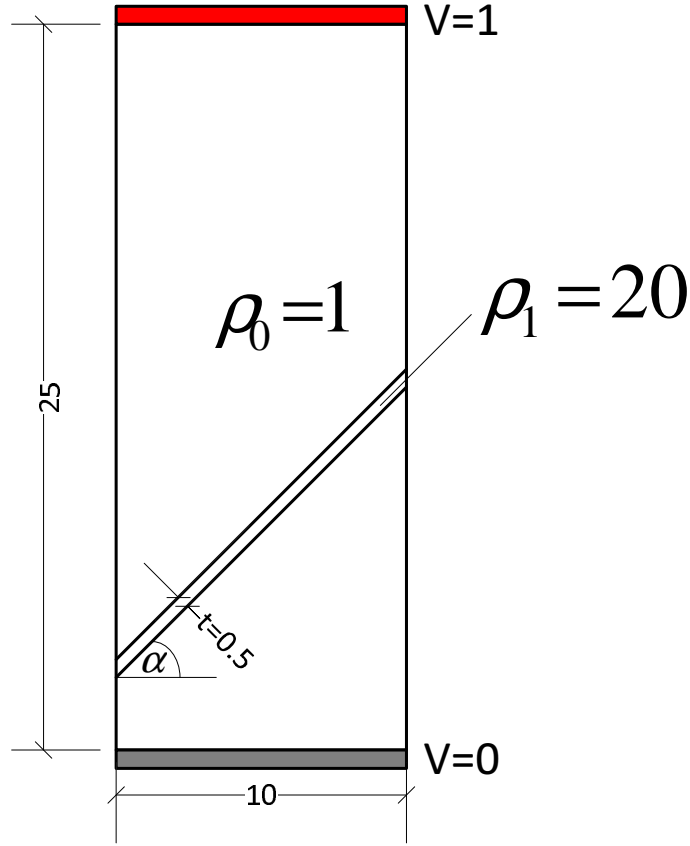


Figure 4.15: Experiment I: Resistance extraction case for a very thin non-rectilinear conductor.

4.5.1.1 FDM with Cartesian grid

As shown in Fig. 4.16, we use very dense Cartesian grid to form the high density staircases to approximate the non-orthogonal interface. From Table 4.2, we found that the accuracy is very low when the mesh is coarse. It requires dense mesh grid to compensate which brings much more computational cost. Especially, when the rotation angle becomes larger, it needs even higher mesh density to achieve the same accuracy.

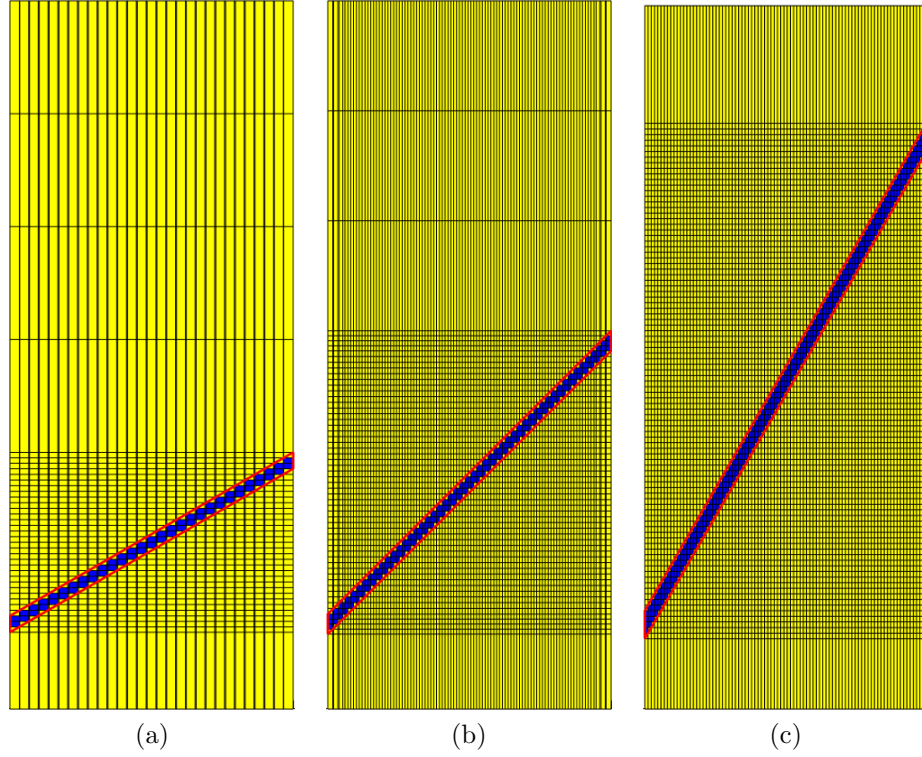


Figure 4.16: Cartesian grid fitting the stripe shaped conductor with different rotation angle α for case in Fig. 4.15. (a) $\alpha = 30^\circ$; (b) $\alpha = 45^\circ$; (c) $\alpha = 60^\circ$.

4.5.1.2 TPFA with Voronoi grid

In order to apply TPFA, we need to place fixed voronoi cell centers to ensure the voronoi cell boundary will fit the rotated stripe domain. Since the potential changes faster along the interface, it is reasonable to increase the mesh density in those sensitive regions. Examples can be found in Fig. 4.17.

Compared with FDM, TPFA are able to get very accurate result with fewer computational cost. For example, to achieve the same accuracy when rotation angle is 45° , FDM uses more than 200 times number of unknowns and more than 150 iterations to calculate. Although it is much more simple and efficient to use TPFA, there are two problems with the voronoi mesh for TPFA. When pre-fixed points are

Table 4.2: Resistance extraction for Experiment I with FDM.

α	dof	R	err(%)	iter
30	588	2.52	-24.53	19
	2204	2.82	-15.52	35
	8120	3.09	-7.51	49
	31088	3.22	-3.58	74
45	1600	2.54	-20.96	32
	5900	2.86	-10.87	54
	22400	3.03	-5.71	95
	87600	3.12	-2.82	153
60	4400	2.68	-12.52	49
	16530	2.86	-6.53	106
	64356	2.96	-3.29	177
	255348	3.01	-1.66	245

Table 4.3: Resistance extraction for Experiment I with TPFA.

α	Voronoi Grid				Voronoi Grid with quad-tree LGR			
	dof	R	err(%)	iter	dof	R	err(%)	iter
30	335	3.35	0.17	22	439	3.35	0.17	29
45	343	3.22	0.19	22	449	3.22	0.13	28
60	365	3.07	0.31	23	526	3.07	0.15	33

utilized to approximate the non-orthogonal interface, it is impossible to make the voronoi cells to completely fit the boundary at the corner point, or the intersection point of two lines, even with highly adjusted parameters. For example, the intersection of the rotated parts and the outer boundary does not fit the original interface line in Fig .4.17(c) and Fig .4.17(f). Another problem is that it is a one-time mesh which needs to update even for a tiny change.

4.5.1.3 MPFA with Triangular grid

To apply MPFA, we employ the package triangle [55] to generate triangular mesh as shown in Fig. 4.18. In order to estimate the performance of MPFA, we compare

the simulation result with the traditional FEM. The result in Table 4.4 shows both of them converge fast and can achieve high accuracy with small dof compared with FDM. It is worth noting that for the same triangular mesh, the number of nodes is roughly half of the number of cells, thus the dof of FEM is less than the dof of MPFA.

Table 4.4: Resistance extraction for Experiment I with MPFA and FEM.

α	MPFA				FEM			
	dof	R	err(%)	iter	dof	R	err(%)	iter
30	94	3.35	0.30	11	60	3.35	0.13	13
	399	3.35	0.30	16	232	3.34	-0.07	23
	773	3.35	0.27	28	428	3.34	-0.10	34
	38796	3.34	0.13	189	19696	3.34	0.00	219
45	127	3.23	0.29	13	77	3.19	-0.80	14
	424	3.23	0.29	21	244	3.22	-0.10	23
	782	3.23	0.29	29	434	3.21	-0.29	33
	38716	3.22	0.06	189	19664	3.22	-0.00	214
60	140	3.07	0.21	14	84	3.06	-0.03	15
	430	3.07	0.18	23	247	3.05	-0.46	23
	800	3.07	0.18	29	444	3.05	-0.31	32
	38875	3.06	0.09	191	19740	3.06	-0.00	213

4.5.1.4 IFEM with Cartesian grid

The original IFEM requires that at most one interface line segment cutting through an element, and the interface line is smooth enough within the interface elements. However, since the stripe shaped conductor is very thin, it is possible that there are two disjoint line segments cutting through some interface elements when the Cartesian grid is coarse. We can extend the non-conforming IFEM for two disjoint interface line segments. For example, in Fig. 4.19, there are two line segments

cutting through the element $\triangle ABC$. There will be three basis functions for three sub-domains respectively,

$$\begin{aligned} V^1(x, y) &= a_0 + a_1x + a_2y, & (x, y) \in \triangle ADE \\ V^2(x, y) &= b_0 + b_1x + b_2y, & (x, y) \in \square DFGE \\ V^3(x, y) &= c_0 + c_1x + c_2y. & (x, y) \in \square FBCG \end{aligned}$$

The potential and normal flux continuity at the edge DE which has the slope α_{DE} can be written as

$$\begin{aligned} a_0 + a_1x_4 + a_2y_4 &= b_0 + b_1x_4 + b_2y_4, \\ a_0 + a_1x_5 + a_2y_5 &= b_0 + b_1x_5 + b_2y_5, \\ \sigma_1(a_1\alpha_{DE} - a_2) &= \sigma_2(b_1\alpha_{DE} - b_2). \end{aligned}$$

Similar equations for the edge FG which has the slope α_{FG} ,

$$\begin{aligned} b_0 + b_1x_6 + b_2y_6 &= c_0 + c_1x_6 + c_2y_6, \\ b_0 + b_1x_7 + b_2y_7 &= c_0 + c_1x_7 + c_2y_7, \\ \sigma_2(b_1\alpha_{FG} - b_2) &= \sigma_3(c_1\alpha_{FG} - c_2). \end{aligned}$$

Thus, the local linear system is

$$\begin{bmatrix}
 1 & x_1 & y_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_2 & y_2 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_3 & y_3 \\
 1 & x_4 & y_4 & -1 & -x_4 & -y_4 & 0 & 0 & 0 \\
 1 & x_5 & y_5 & -1 & -x_5 & -y_5 & 0 & 0 & 0 \\
 0 & \alpha_{DE} & -1 & 0 & -\frac{\sigma_2}{\sigma_1}\alpha_{DE} & \frac{\sigma_2}{\sigma_1} & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & x_6 & y_6 & -1 & -x_6 & -y_6 \\
 0 & 0 & 0 & 1 & x_7 & y_7 & -1 & -x_7 & -y_7 \\
 0 & 0 & 0 & 0 & \alpha_{FG} & -1 & 0 & -\frac{\sigma_3}{\sigma_2}\alpha_{FG} & \frac{\sigma_3}{\sigma_2}
 \end{bmatrix}
 \begin{bmatrix}
 a_0 \\
 a_1 \\
 a_2 \\
 b_0 \\
 b_1 \\
 b_2 \\
 c_0 \\
 c_1 \\
 c_2
 \end{bmatrix}
 =
 \begin{bmatrix}
 V(A) \\
 V(B) \\
 V(C) \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{bmatrix}.$$

Suppose there are more than two interface lines, whenever there is another disjoint line segment added to the local system, there will be one more basis function for the new sub-domain which has three unknown parameters. While three equations determined by the potential continuity at the two intersection points and flux continuity at the normal direction of the edge, will make the updated local system still solvable.

In the experiment for non-rectilinear conductor, we employ very coarse mesh grid as shown in Fig. 4.20. All the results in Table. 4.5 are very accurate, even when the number of grid point is less than 10. The advantage of IFEM over FDM, FVM and FEM is obvious. It does not require any specific technique to generate complicated mesh to fit the non-orthogonal interface or update mesh for any small changes, which greatly saves the computational cost. In addition, since the matrix construction for non-interface elements still maintain the FDM five-point stencil, it only requires solving local system for non-interface elements. Thus, it is much more efficient than

other approaches.

Table 4.5: Resistance extraction for Experiment I with IFEM.

α	dof	R	err(%)	iter
30	1	3.32	-0.56	0
	9	3.34	0.00	1
	25	3.34	-0.03	6
	81	3.34	-0.07	11
	289	3.34	-0.03	22
45	1	3.17	-1.49	0
	9	3.08	-4.28	1
	25	3.21	-0.16	6
	81	3.22	0.06	11
	289	3.22	0.00	21
60	1	0.98	-2.89	0
	9	3.15	2.71	1
	25	3.04	-0.67	7
	81	3.09	0.80	12
	289	3.08	0.43	21

4.5.2 Resistance Extraction of FinFET Contact

The resistivity between the boundary of FinFET gate and contact is anisotropic, which means that the resistivity in the normal direction is different from that in the tangent direction as shown in Fig. 4.21(a). In order to take this property into consideration, we create a very thin layer with different resistivity. All dimensional detail is depicted in Fig. 4.21(b). The half circle curve is approximated by thirty two equal length segments. We did experiments for the case $\rho_1/\rho_2 = 20$ and $\rho_1/\rho_2 = 200$. The reference result from FEM solver is $R = 2.35\Omega$ for the former case, and $R = 11.14\Omega$ for the latter case with $dof = 155903$.

Table 4.6: Resistance extraction for Experiment II with FDM.

cases	dof	R	err(%)	iter	runtime(s)
$\rho_1/\rho_2 = 20$ $R_{ref} = 2.35\Omega$	506	1.69	-27.82	19	0.41
	1720	2.05	-12.63	30	1.04
	6308	2.20	-6.04	47	3.49
	24124	2.27	-3.20	78	14.39
	94316	2.32	-1.00	127	124.25
	372940	2.35	0.31	214	2118.5
$\rho_1/\rho_2 = 200$ $R_{ref} = 11.14\Omega$	506	1.96	-82.37	20	0.36
	1720	7.84	-29.62	32	1.00
	6308	9.66	-13.24	62	3.46
	24124	10.40	-6.65	121	14.77
	94316	10.78	-3.23	221	132.95
	372940	10.95	-1.64	377	2029.76

4.5.2.1 FDM with Cartesian grid

It requires very dense Cartesian grid to approximate the non-orthogonal boundary as usual, which is demonstrated in Fig. 4.22. Though coarse mesh is used for domains without non-orthogonal interface, the dof is still very large to ensure accuracy. The results in Table. 4.6 show poor results for coarse mesh even though it converges fast. As the ratio ρ_1/ρ_2 becomes large, it requires even more grid points to guarantee the accuracy, which leads to much more computational costs. Therefore, as the resistivity contrast along the non-orthogonal interface becomes larger, the performance of FDM is even worse.

4.5.2.2 TPFA with Voronoi grid

Fig. 4.23 displays Voronoi grid examples with and without local grid refinement. Since there are multiple line segments to form the curve shape, the Voronoi cells around the intersection point of the line segments can not fully match the boundary lines. Therefore, the Voronoi cells boundaries approximate the non-orthogonal inter-

Table 4.7: Resistance extraction for Experiment II with TPFA.

cases	Voronoi Grid					Voronoi Grid with quad-tree LGR				
	dof	R	err(%)	iter	runtime(s)	dof	R	err(%)	iter	runtime(s)
$\rho_1/\rho_2 = 20$	473	2.31	-1.48	27	0.21	619	2.32	-1.04	30	0.23
$\rho_1/\rho_2 = 200$	473	11.16	0.22	27	0.18	619	11.12	-0.11	31	0.22

faces with trivial accuracy loss. When local grid refinement is included, greater mesh density is applied for interface domains while lower mesh density for other domains, which is much more efficient.

4.5.2.3 MPFA with Triangular grid

With the triangular mesh in Fig.4.24 generated by triangle package, we did experiments with both MPFA and FEM, and compare both accuracy, ICCG iteration times as well as run times in Table. 4.8. Both of them can achieve high accuracy with small number of grid points which is much more efficient than FDM. Their iteration number is comparable. The simulation with FEM is much faster than MPFA when dof is small while becomes slower when dof is large. The reason is it is expensive to construct the local matrix for a single node in MPFA than constructing the local matrix for a single element in FEM, while the number of nodes is around half of the number of cells for the same triangular mesh. As the dof increases, the run time becomes much more sensitive to the total assembling times of the local matrix.

4.5.2.4 IFEM with Cartesian grid

In this experiment, we adopts Cartesian grid for non-interface elements, while Cartesian triangular grid for interface elements, as shown in 4.25. Traditional FDM will be applied to the rectangle cells, while IFEM is limited to those interface cells. It is clear that the number of interface cells is proportional to the dimension of the non-orthogonal interfaces, which is very small. Thus, the cost to solve the local

Table 4.8: Resistance extraction for Experiment II with MPFA and FEM.

cases	MPFA					FEM				
	dof	R	err(%)	iter	runtime(s)	dof	R	err(%)	iter	runtime(s)
$\rho_1/\rho_2 = 20$	336	2.44	4.00	19	0.41	182	2.27	-3.36	20	0.15
	777	2.40	2.13	28	0.68	423	2.29	-2.25	30	0.25
	2521	2.39	0.97	50	1.67	1328	2.32	-0.98	53	0.49
	15477	2.36	0.47	111	8.33	7909	2.34	-0.35	124	2.19
	155055	2.35	0.34	266	83.8	78047	2.34	-0.05	307	98.4
	310200	2.34	-0.12	376	184.50	155903	2.35	0.00	421	401.9
$\rho_1/\rho_2 = 200$	336	11.26	1.13	20	0.44	182	11.04	-0.88	20	0.14
	777	11.20	0.56	30	0.75	423	11.06	-0.66	31	0.20
	2521	11.16	0.22	55	1.82	1328	11.09	-0.44	59	0.41
	15477	11.15	0.11	120	8.33	7909	11.12	-0.22	130	1.95
	155055	11.14	0.06	263	92.94	78047	11.12	-0.11	308	116.25
	310200	11.14	0.03	371	189.00	155903	11.14	0.00	412	401.64

system for non-orthogonal elements is small which avoids the disadvantage of the traditional FEM. From Table. 4.9 and Fig. 4.26, by applying IFEM, high accuracy can be achieved with small number of grid points than FDM, FEM and MPFA. The iteration number is comparable to those with FEM and MPFA, and the run time is smaller than FEM and MPFA with the same accuracy. From Fig. 4.27, the efficiency improvement is even more clear. For instance, when $\rho_1/\rho_2 = 20$, to achieve 1% error for resistance, FDM requires run time of 120s while IFEM only requires less than 1s, which indicates more than 100 times speed up when IFEM is applied. The efficiency improvement is more than 1000 times when $\rho_1/\rho_2 = 200$.

4.6 Conclusion

We introduce FVM and IFEM to solve the resistance extraction problem with non-orthogonal interface. MPFA is similar to FEM, which assembles local matrix for each node instead of element. IFEM applies FDM to non-interface elements and extended FEM to interface element, which combines the advantage of both them.

Experiments have verified that both FVM and IFEM can provide accurate result with small number of unknowns than FDM. The accuracy and efficiency of FVM

Table 4.9: Resistance extraction for Experiment II with IFEM.

cases	dof	R	err(%)	iter	runtime(s)
$\rho_1/\rho_2 = 20$	64	2.26	-3.51	11	0.17
	156	2.31	-1.73	15	0.22
	460	2.32	-1.25	29	0.32
	1591	2.33	-0.79	53	0.71
	5893	2.33	-0.54	102	2.15
	22657	2.34	-0.42	197	10.3
$\rho_1/\rho_2 = 200$	64	9.37	-15.84	12	0.18
	156	10.91	-1.97	15	0.23
	460	11.09	-0.44	29	0.33
	1591	11.11	-0.22	54	0.73
	5893	11.12	-0.11	103	2.22
	22657	11.12	-0.11	197	10.8

is comparable to FEM, with the same non-cartesian mesh generation requirement. IFEM provides better efficiency with the simple Cartesian mesh which greatly reduces the mesh generation costs, especially when few geometric changes are made.

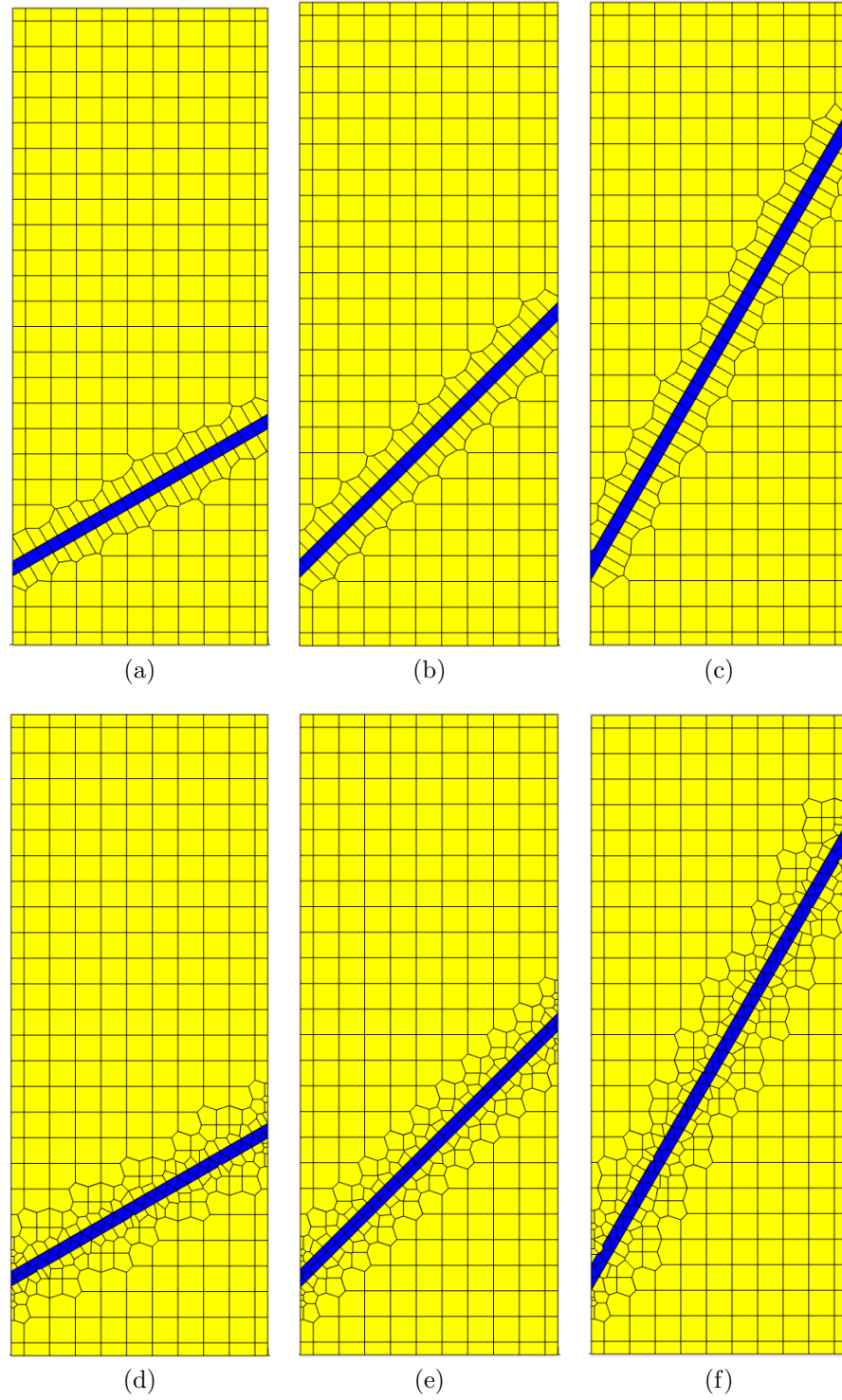


Figure 4.17: Voronoi grid fitting the stripe shaped conductor with different rotation angle α for case in Fig. 4.15. (a) $\alpha = 30^\circ$ without LGR; (b) $\alpha = 45^\circ$ without LGR; (c) $\alpha = 60^\circ$ without LGR; (d) $\alpha = 30^\circ$ with quad-tree LGR; (e) $\alpha = 45^\circ$ with quad-tree LGR; (f) $\alpha = 60^\circ$ with quad-tree LGR.

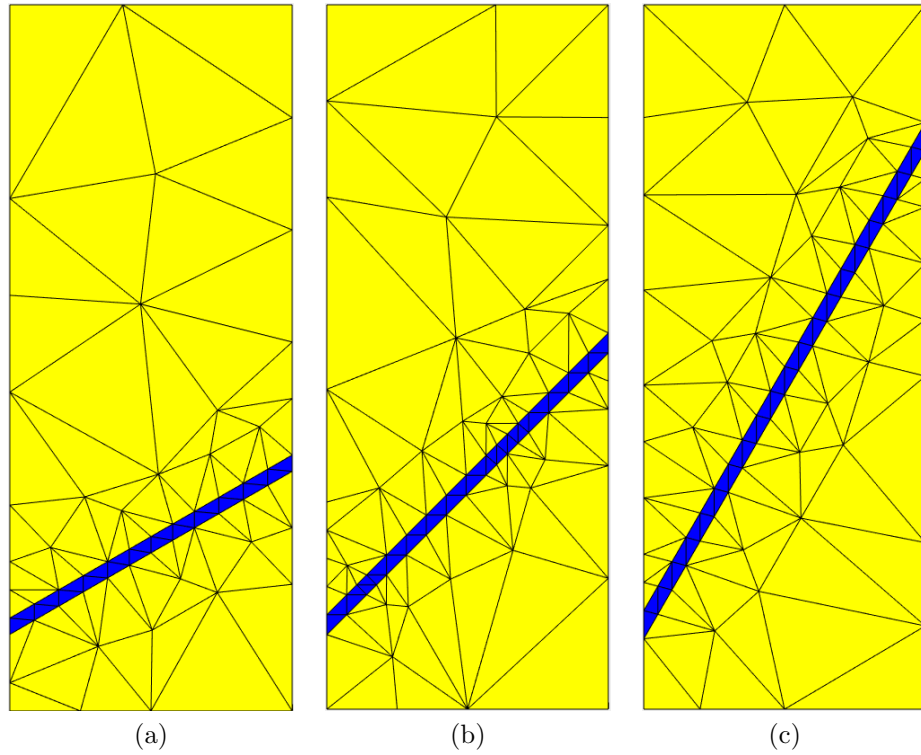


Figure 4.18: Triangular mesh fitting the stripe shaped conductor with different rotation angle α for case in Fig. 4.15. (a) $\alpha = 30^\circ$; (b) $\alpha = 45^\circ$; (c) $\alpha = 60^\circ$.

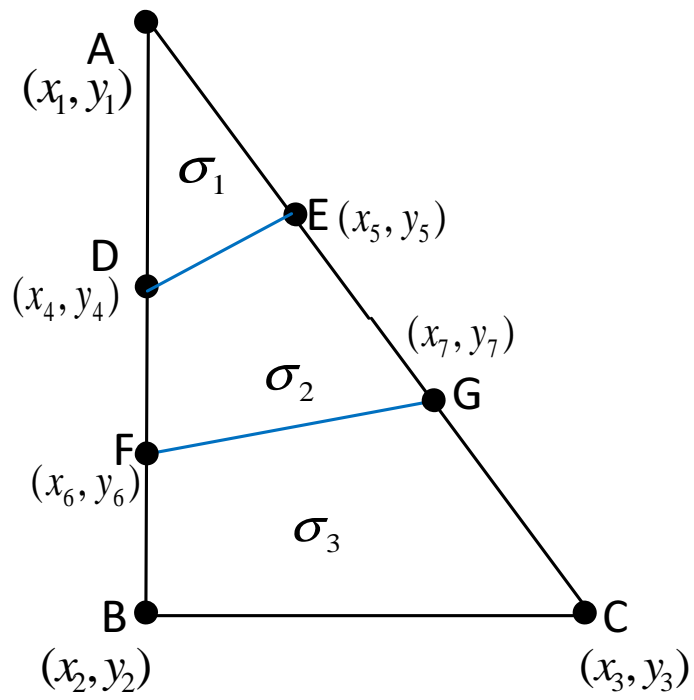


Figure 4.19: Cartesian triangle with two disjoint interface lines cutting through.

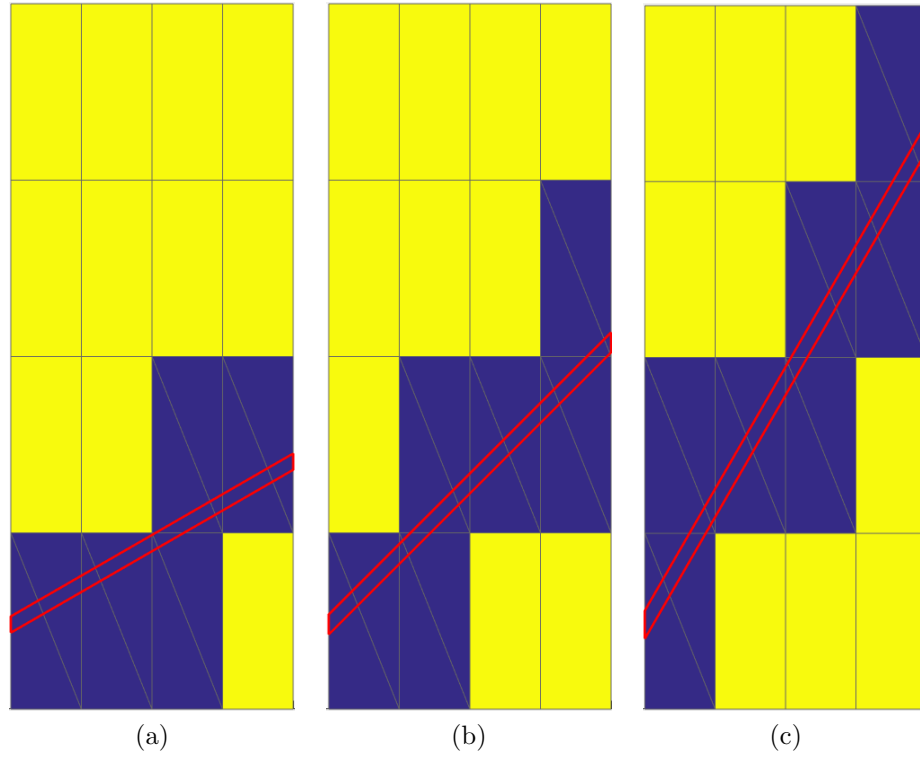


Figure 4.20: Cartesian mesh for non-interface elements and Cartesian triangular mesh for interface elements for case in Fig. 4.15. (a) $\alpha = 30^\circ$; (b) $\alpha = 45^\circ$; (c) $\alpha = 60^\circ$. All interface elements are marked with blue color while the rest elements are labeled with yellow color.

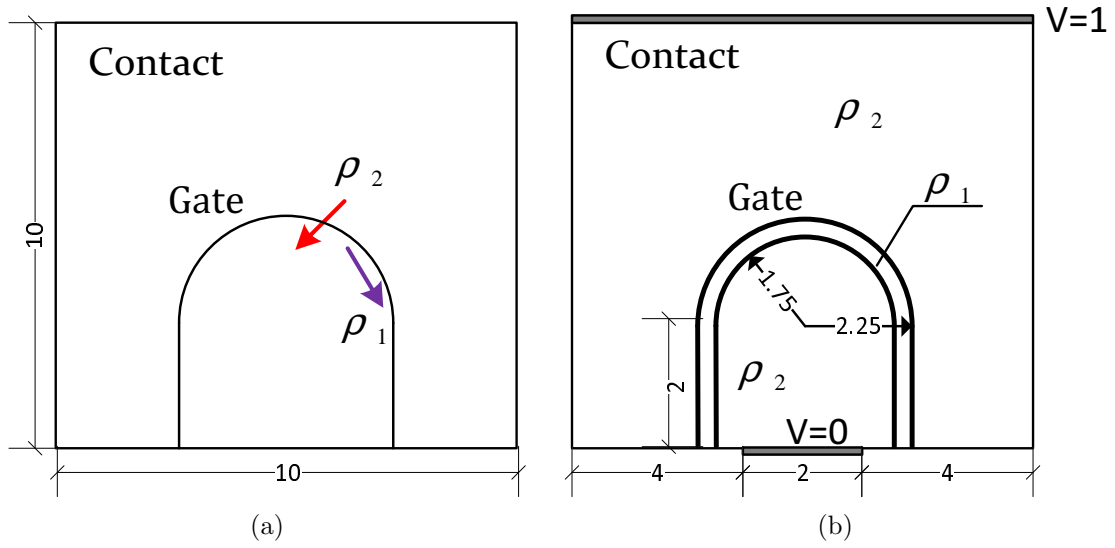


Figure 4.21: Experiment II: Resistivity at the boundary between FinFET gate and contact is anisotropic. (a) different resistivity in the normal direction and tangent direction of the boundary (b) equivalent model with a very thin layer at the boundary.

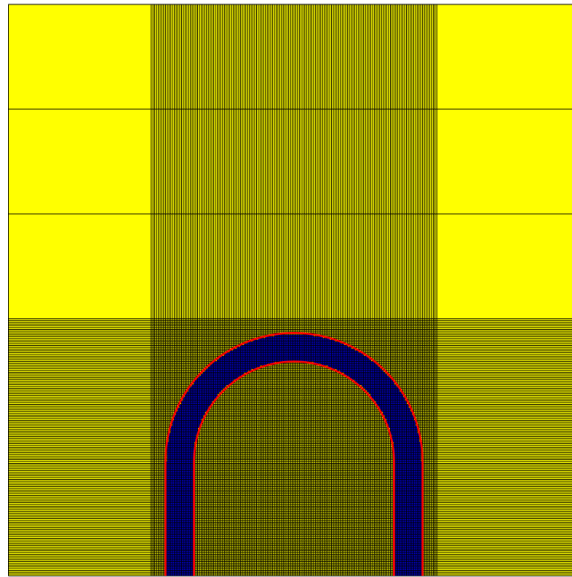


Figure 4.22: Cartesian grid fitting the thin layer between FinFET gate and contact in Fig. 4.21.

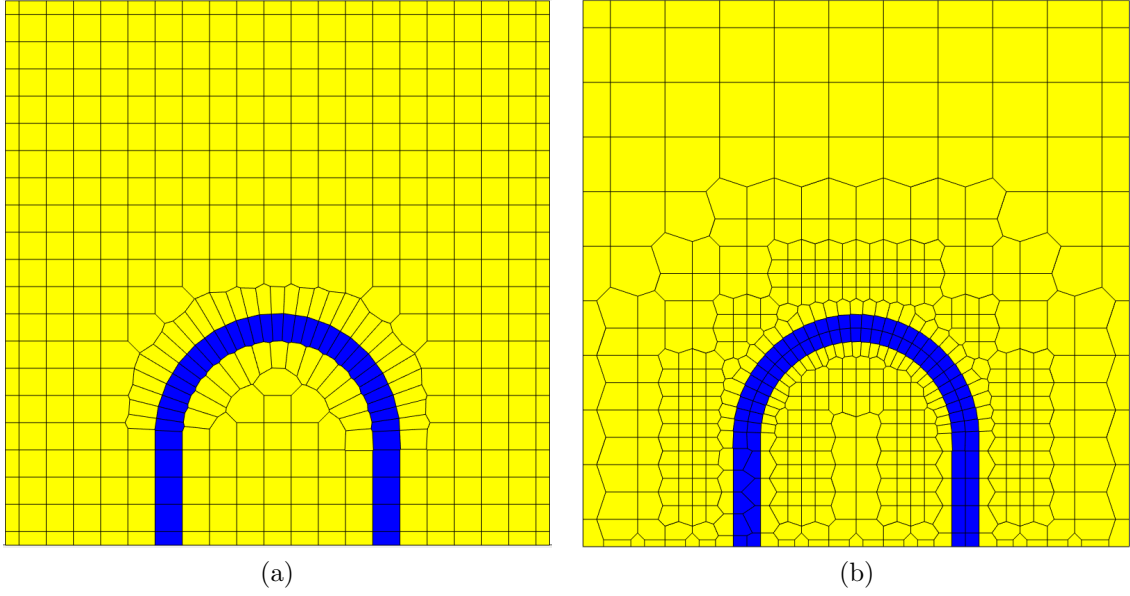


Figure 4.23: Voronoi mesh fitting the thin layer between FinFET gate and contact in Fig. 4.21. (a) without local grid refinement (b) with multi-level quad-tree local grid refinement.

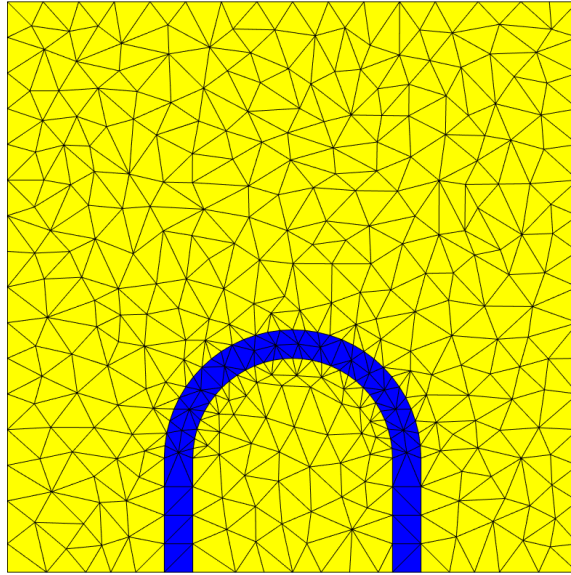


Figure 4.24: Triangular mesh fitting the thin layer between FinFET gate and contact in Fig. 4.21.

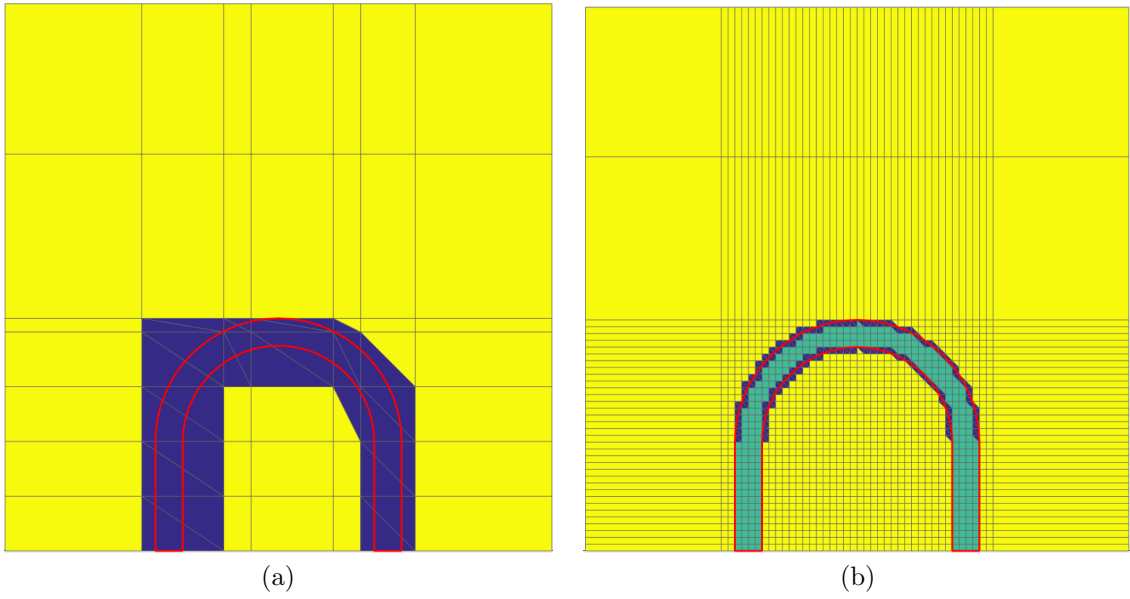


Figure 4.25: Cartesian mesh for non-interface elements and Cartesian triangular mesh for interface elements for case in Fig. 4.21. (a)coarse mesh with two lines cutting through elements (b)dense mesh. All interface elements are marked with blue color. Non-interface elements outside of the thin layer are labeled with yellow color, while non-interface elements inside of the thin layer are labeled with green color.

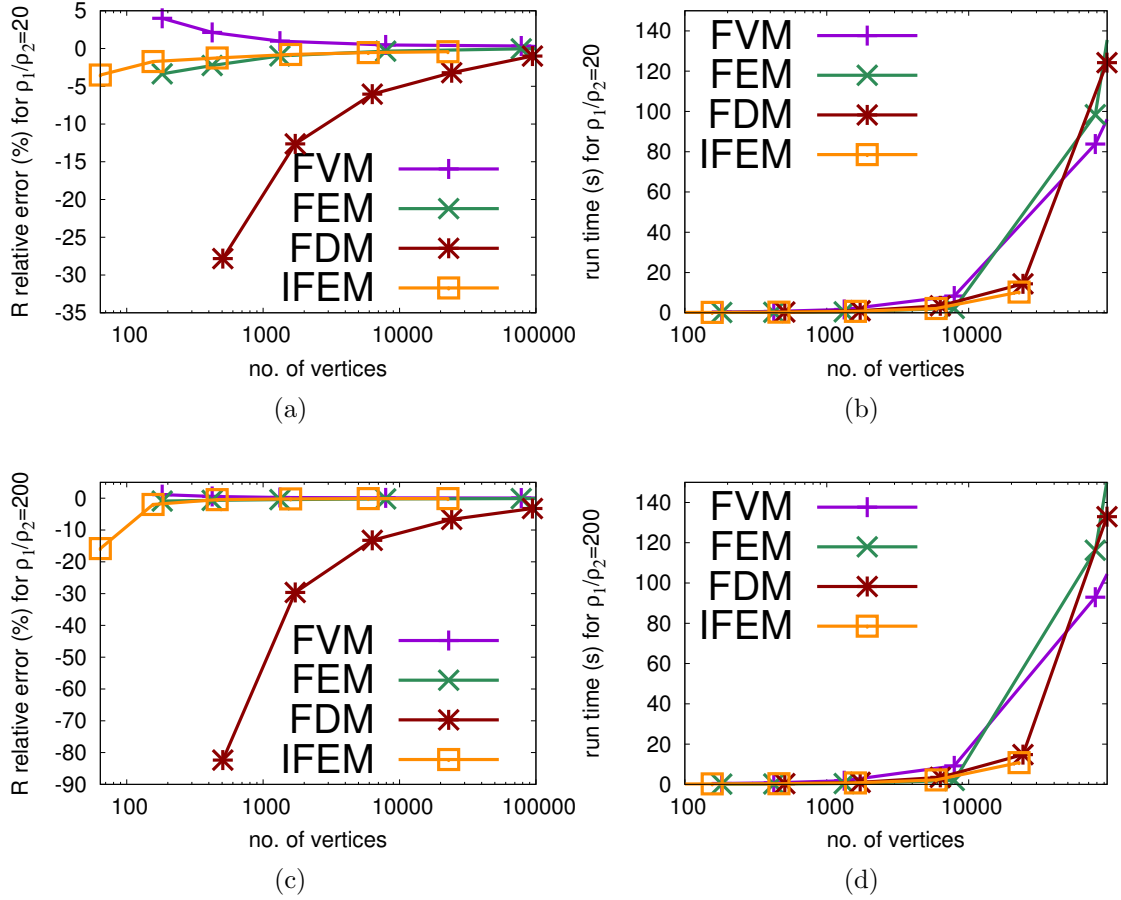


Figure 4.26: Accuracy convergence and run time comparison for Experiment II.

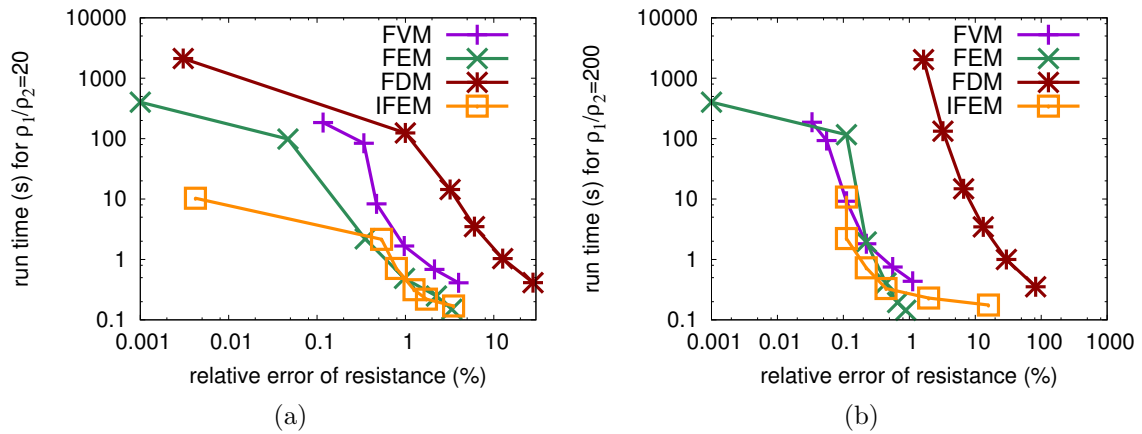


Figure 4.27: Run time as a function of relative error for resistance in Experiment II.

5. CONCLUSION

In this dissertation, we present several practical techniques for parasitic extraction of advanced integrated circuits.

We first tackle the problem of capacitance extraction for interconnect with sensitive information hided. We propose the macro model around any region that foundries or IP vendors wish to hide information, which allows accurate capacitance extraction inside and outside of the region. The novelty of this approach is that the geometrical and dielectric constant information within the sensitive region is irreversible, which can effectively protect trade secrets for foundries and vendors. Experimental results demonstrate that there is no accuracy loss when the mesh density inside and outside of the sensitive region are the same, while the error is less than 1% when the mesh density is inconsistent.

The second problem we investigate is to decrease the truncation error introduced by the Neumann boundary condition. We propose a class of absorbing boundary conditions, which when implemented, significantly reduce the distortion of the field at the numerical boundary and consequently throughout the numerical region. The absorbing boundary condition we propose will allow the field throughout the numerical region to behave as though there is no numerical boundary, accurately mimicking the fields in an actual region. As a result, the size of the numerical region can significantly reduced, which in turn reduces the run time without sacrificing accuracy. Experimental results for capacitance extraction with interconnects in multi-layer dielectrics and SOI show the proposed methods have three to ten times faster run times for the same accuracy.

The third problem we focus on is to improve the accuracy and efficiency of resis-

tance extraction for FinFET with non-orthogonal conductivity interface. The bottleneck for the application of FDM is that it requires very dense Cartesian grid to approximate the non-orthogonal interface. Though FEM is an applicable approach, the costs to generate proper triangular mesh is non-trivial. Besides, it is expensive to solve the local system. We introduce both FVM and IFEM to resolve this problem. The performance of FVM is comparable to FEM, which faces the similar dilemma of mesh generation costs. IFEM combines the advantage of both FDM and FEM, which maintains simple stencil for regions without non-orthogonal interface, while solving the local system for the grid cell elements with interface lines. It has better accuracy than all other approaches, and avoids the high mesh generation costs effectively.

REFERENCES

- [1] M. Bohr and K. Mistry, “Intels revolutionary 22 nm transistor technology,” *Intel website*, 2011.
- [2] K. Chow and C. Relyea, “Calibre xACT3D - no compromise extraction for advanced transistor level design.” Mentor Graphics white paper, June 2010.
- [3] O. Shah and M. Koshy, “StarRC custom Rapid3D extraction next generation high performance 3D fast field solver.” Synopsys white paper, June 2010.
- [4] R. Iverson and Y. Le Coz, “User guide for quickcap version 3.0,” *Random Logic Corporation*, 1999.
- [5] Star-RCXT Parasitic extraction DataSheet. Synopsys, 2006.
- [6] Assura Physical Verification (Design rule checking and layout vs. schematic verification) DataSheet. Cadence Design Systems, 2012.
- [7] Calibre xRC DataSheet. Mentor Graphics, 2011.
- [8] E. A. Dengi and R. A. Rohrer, “Hierarchical 2-D field solution for capacitance extraction for VLSI interconnect modeling,” in *Proceedings of the 34th annual Design Automation Conference*, pp. 127–132, ACM, 1997.
- [9] E. A. Dengi and R. A. Rohrer, “Boundary element method macromodels for 2-D hierarchical capacitance extraction,” in *Proc. ACM/IEEE Design Automation Conf.*, pp. 218–223, 1998.
- [10] W. Shi and W. Qiu, “Encrypted profiles for parasitic extraction.” Patent US 8499263B1, filing date Mar. 29th, 2012, grant publication date July. 30th, 2013.
- [11] M. N. Sadiku, *Numerical techniques in electromagnetics*. CRC press, 2000.

- [12] O. C. Zienkiewicz, R. L. Taylor, and J. Zhu, *The finite element method: its basis and fundamentals*. Oxford, U.K.: Butterworth-Heinemann, 6th ed., 2005.
- [13] Y. Le Coz and R. Iverson, “A stochastic algorithm for high speed capacitance extraction in integrated circuits,” *Solid-State Electronics*, vol. 35, no. 7, pp. 1005–1012, 1992.
- [14] N. J. Higham, *Accuracy and stability of numerical algorithms*. Siam, 2002.
- [15] D. R. Kincaid and E. W. Cheney, *Numerical analysis: mathematics of scientific computing*, vol. 2. American Mathematical Soc., 2002.
- [16] R. Haberman, *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*. Pearson Higher Ed, 2012.
- [17] B. Engquist and A. Majda, “Absorbing boundary conditions for numerical simulation of waves,” *Proceedings of the National Academy of Sciences*, vol. 74, no. 5, pp. 1765–1766, 1977.
- [18] J.-P. Berenger, “A perfectly matched layer for the absorption of electromagnetic waves,” *Journal of computational physics*, vol. 114, no. 2, pp. 185–200, 1994.
- [19] R. K. Gordon and S. H. Fook, “A finite difference approach that employs an asymptotic boundary condition on a rectangular outer boundary for modeling two-dimensional transmission line structures,” *IEEE Transactions on microwave theory and techniques*, vol. 41, no. 8, pp. 1280–1286, 1993.
- [20] D. Ioan, G. Ciuprina, and M. Radulescu, “Absorbing boundary conditions for compact modeling of on-chip passive structures,” *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 25, no. 3, pp. 652–659, 2006.

- [21] T. Hagstrom and H. Keller, “Asymptotic boundary conditions and numerical methods for nonlinear elliptic problems on unbounded domains,” *Mathematics of computation*, vol. 48, no. 178, pp. 449–470, 1987.
- [22] L. N. Trefethen, “Finite difference and spectral methods for ordinary and partial differential equations,” 1996.
- [23] P. Bekal, “Analysis and modeling of parasitic capacitances in advanced nanoscale devices,” Master’s thesis, Texas A & M University, 2012.
- [24] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, vol. 55. Courier Corporation, 1964.
- [25] P. Moon and D. E. Spencer, *Field theory handbook: including coordinate systems, differential equations and their solutions*. Springer, 2012.
- [26] L. Greengard, *The rapid evaluation of potential fields in particle systems*. MIT press, 1988.
- [27] R. Courant and D. Hilbert, *Methods of mathematical physics*, vol. 1. CUP Archive, 1965.
- [28] I. Bardi, O. Biro, and K. Preis, “Perfectly matched layers in static fields,” *IEEE transactions on magnetics*, vol. 34, no. 5, pp. 2433–2436, 1998.
- [29] L. Dedek, J. Dedkova, and J. Valsa, “Optimization of perfectly matched layer for laplace’s equation,” *IEEE transactions on magnetics*, vol. 38, no. 2, pp. 501–504, 2002.
- [30] B. S. Guru and H. R. Hiziroglu, *Electromagnetic field theory fundamentals*. Cambridge university press, 2004.

- [31] J. N. Jere and Y. Le Coz, “An improved floating-random-walk algorithm for solving the multi-dielectric dirichlet problem,” *IEEE transactions on microwave theory and techniques*, vol. 41, no. 2, pp. 325–329, 1993.
- [32] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, “Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 35, no. 3, p. 22, 2008.
- [33] A. Seidl, H. Klose, M. Svoboda, J. Oberndorfer, and W. Rosner, “Capcal-a 3-d capacitance solver for support of cad systems,” *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 7, no. 5, pp. 549–556, 1988.
- [34] T.-Y. Chou and Z. J. Cendes, “Capacitance calculation of ic packages using the finite element method and planes of symmetry,” *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 13, no. 9, pp. 1159–1166, 1994.
- [35] C. Hirsch, *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*. Butterworth-Heinemann, 2007.
- [36] I. Aavatsmark, “An introduction to multipoint flux approximations for quadrilateral grids,” *Computational Geosciences*, vol. 6, no. 3-4, pp. 405–432, 2002.
- [37] R. Eymard, T. Gallouët, and R. Herbin, “Finite volume methods,” *Handbook of numerical analysis*, vol. 7, pp. 713–1018, 2000.
- [38] B. Diskin, J. L. Thomas, E. J. Nielsen, H. Nishikawa, and J. A. White, “Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes,” *AIAA journal*, vol. 48, no. 7, pp. 1326–1338, 2010.

- [39] J. Y. Murthy and S. Mathur, “Numerical methods in heat, mass, and momentum transfer,” *School of Mechanical Engineering, Purdue University*, 2002.
- [40] D. Holmes and S. Connell, “Solution of the 2d navier-stokes equations on unstructured adaptive grids,” in *9th Computational Fluid Dynamics Conference*, p. 1932, 1989.
- [41] T. BARTH, “A 3-d upwind euler solver for unstructured meshes,” in *10th Computational Fluid Dynamics Conference*, p. 1548, 1991.
- [42] D. Mavriplis, “Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes,” in *16th AIAA Computational Fluid Dynamics Conference*, p. 3986, 2003.
- [43] J. Morel, J. Dendy, M. L. Hall, and S. W. White, “A cell-centered lagrangian-mesh diffusion differencing scheme,” *Journal of Computational Physics*, vol. 103, no. 2, pp. 286–299, 1992.
- [44] K. Lipnikov, M. Shashkov, and I. Yotov, “Local flux mimetic finite difference methods,” *Numerische Mathematik*, vol. 112, no. 1, pp. 115–152, 2009.
- [45] I. Aavatsmark, T. Barkve, O. Bøe, and T. Mannseth, “Discretization on unstructured grids for inhomogeneous, anisotropic media. part i: Derivation of the methods,” *SIAM Journal on Scientific Computing*, vol. 19, no. 5, pp. 1700–1716, 1998.
- [46] I. Aavatsmark, T. Barkve, O. Bøe, and T. Mannseth, “Discretization on unstructured grids for inhomogeneous, anisotropic media. part ii: Discussion and numerical results,” *SIAM Journal on Scientific Computing*, vol. 19, no. 5, pp. 1717–1736, 1998.

- [47] J. Breil and P.-H. Maire, “A cell-centered diffusion scheme on two-dimensional unstructured meshes,” *Journal of Computational Physics*, vol. 224, no. 2, pp. 785–823, 2007.
- [48] H. A. Friis, M. G. Edwards, and J. Mykkeltveit, “Symmetric positive definite flux-continuous full-tensor finite-volume schemes on unstructured cell-centered triangular grids,” *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 1192–1220, 2008.
- [49] I. D. Mishev, “Finite volume methods on voronoi meshes,” *Numerical methods for Partial Differential equations*, vol. 14, no. 2, pp. 193–212, 1998.
- [50] C. L. Palagi, K. Aziz, *et al.*, “Use of voronoi grid in reservoir simulation,” *SPE Advanced Technology Series*, vol. 2, no. 02, pp. 69–77, 1994.
- [51] R. Liska, M. Shashkov, and V. Ganzha, “Analysis and optimization of inner products for mimetic finite difference methods on a triangular grid,” *Mathematics and Computers in Simulation*, vol. 67, no. 1, pp. 55–66, 2004.
- [52] R. Klausen, F. Radu, and G. Eigestad, “Convergence of mpfa on triangulations and for richards’ equation,” *International journal for numerical methods in fluids*, vol. 58, no. 12, pp. 1327–1351, 2008.
- [53] Z. Li, T. Lin, and X. Wu, “New cartesian grid methods for interface problems using the finite element formulation,” *Numerische Mathematik*, vol. 96, no. 1, pp. 61–98, 2003.
- [54] Z. Li, “An overview of the immersed interface method and its applications,” *Taiwanese journal of mathematics*, pp. 1–49, 2003.
- [55] J. R. Shewchuk, “Triangle: Engineering a 2d quality mesh generator and delaunay triangulator,” in *Applied computational geometry towards geometric engi-*

neering, pp. 203–222, Springer, 1996.